Recherche d'une représentation des données efficace pour la fouille des grandes bases de données

Marc Boullé

École Nationale Supérieure des Télécommunications

ÉCOLE DOCTORALE : EDITE

THÈSE

présentée par

Marc BOULLÉ

et soutenue

le 24 septembre 2007

en vue de l'obtention du

Grade de Docteur de
l'École Nationale Supérieure des Télécommunications

spécialité : Informatique

Recherche d'une représentation des données efficace pour la fouille des grandes bases de données

MEMBRES du JURY

Isabelle Guyon	**Société ClopiNet**	**Rapporteur**
Christian Robert	**Ceremade - Université Paris-Dauphine**	**Rapporteur**
Eric Moulines	**TSI - ENST**	**Directeur**
Fabrice Clérot	**France Télécom R&D de Lannion**	**Co-encadrant**
Michèle Sebag	**LRI - Université Paris-Sud**	**Examinateur**
Djamel Zighed	**ERIC - Université Lumière Lyon 2**	**Examinateur**

École Nationale Supérieure des Télécommunications

ÉCOLE DOCTORALE : EDITE

THÈSE

présentée par

Marc BOULLÉ

et soutenue

le 24 septembre 2007

en vue de l'obtention du

Grade de Docteur de
l'École Nationale Supérieure des Télécommunications

spécialité : Informatique

Recherche d'une représentation des données efficace pour la fouille des grandes bases de données

MEMBRES du JURY

Isabelle Guyon	**Société ClopiNet**	**Rapporteur**
Christian Robert	**Ceremade - Université Paris-Dauphine**	**Rapporteur**
Eric Moulines	**TSI - ENST**	**Directeur**
Fabrice Clérot	**France Télécom R&D de Lannion**	**Co-encadrant**
Michèle Sebag	**LRI - Université Paris-Sud**	**Examinateur**
Djamel Zighed	**ERIC - Université Lumière Lyon 2**	**Examinateur**

Remerciements

Je voudrais remercier ici sincèrement tous ceux qui m'ont aidé pendant mes années de thèse.

J'exprime d'abord ma profonde gratitude à Fabrice CLÉROT, qui par sa culture scientifique, sa disponibilité et son ouverture d'esprit, m'a procuré un cadre de travail tout autant qu'un espace de liberté hors normes.

Je suis très reconnaissant à Eric MOULINES, Professeur à l'ENST, d'avoir dirigé ma thèse et apporté ses connaissances et compétences lors d'entrevues particulièrement stimulantes.

Je remercie vivement Isabelle GUYON et Christian ROBERT, qui ont accepté de rapporter ce travail, m'ont reçu longuement pour en discuter et m'ont éclairé par leurs remarques approfondies.

Je tiens également à remercier Michèle SEBAG et Djamel ZIGHED d'avoir accepté de participer à ce jury et d'avoir émis au cours de la soutenance des remarques et suggestions qui alimenteront la suite de mes travaux.

Le présent travail a été effectué à France Télécom R&D Lannion au sein de l'équipe Traitement Statistique de l'Information. J'ai ici bénéficié d'un environnement de travail extrêmement favorable, avec des interactions constructives sur la théorie et la pratique de la fouille de données en entreprise ainsi que de nombreux échanges chaleureux. Merci à tous les membres de l'équipe que je ne peux citer nommémemt, et plus généralement à tous mes collègues de France Télécom avec qui j'ai eu le plaisir de travailler. Merci spécialement à Carine HUE et Sylvain FERRANDIZ qui m'ont accompagné dans ce travail.

Enfin, un grand MERCI à toute ma famille, ma femme Kitty et mes enfants Nicolas, Olivier et Benjamin.

A Kitty, Nicolas, Olivier et Benjamin.

Table des matières

1

Introduction

Sommaire

Le sujet original des travaux décrits dans ce mémoire était

> "Recherche d'une représentation des données efficace pour le Data Mining supervisé dans le cadre des grandes bases de données",

ce qui finalement s'avère proche de l'intitulé actuel. Le contenu de ce mémoire garde ainsi une certaine proximité avec le sujet initial.

Nous présentons dans ce premier chapitre le contexte des travaux, celui de la préparation des données dans le processus Data Mining. Nous précisons ensuite leur objectif, à savoir l'automatisation de la phase de préparation des données. Nous résumons enfin notre contribution, avant d'introduire le plan de ce mémoire.

1.1 Contexte

Après avoir rappelé quelques notions de modélisation statistique et défini le vocabulaire utilisé tout au long de ce mémoire, nous présentons le processus Data Mining, dont nous indiquons les particularités dans un contexte industriel.

1.1.1 Modélisation statistique

La modélisation statistique s'attache à analyser les relations entre des données, disponibles généralement sous la forme d'un ensemble d'*individus* caractérisés par des *variables*. L'individu est l'unité élémentaire de la modélisation statistique. On parle aussi d'instance, d'objet, d'observation, d'enregistrement voire de ligne pour des individus stockés sous la forme d'un tableau de données. Un individu est défini par des *valeurs* correspondant aux variables, aussi appelées attributs ou colonnes. Les données sont ainsi habituellement représentées sous la forme d'un tableau croisé individus * variables.

Les variables sont majoritairement typées en variables *numériques* (ou continues, quantitatives) et variables *catégorielles* (ou catégoriques, qualitatives, nominales). On distingue deux rôles pour les variables : les variables *explicatives* (ou descriptives, en entrée, exogènes) et les variables *à expliquer* (ou cibles, à prédire, en sortie, endogènes).

Selon le type et le rôle des variables, on distingue plusieurs objectifs de modélisation. En présence d'une variable à expliquer Y observée conjointement avec un ensemble $X = \{X_1, X_2, \ldots, X_K\}$ de K variables explicatives, on parle de problème d'*apprentissage supervisé*. Il s'agit d'un problème de *classification supervisée* (ou discrimination) dans le cas de la prédiction d'une variable catégorielle, et d'un problème de *régression* dans le cas de la prédiction d'une variable numérique. En l'absence de variable à expliquer, on parle de problème d'*apprentissage non supervisé* (ou de clustering), et l'objectif consiste généralement à rechercher les classes "naturelles" d'individus "similaires". La modélisation est parfois qualifiée de *prédictive* dans le cas supervisé, et *descriptive* dans le cas non supervisé.

Dans le cas de la modélisation prédictive, on distingue la modélisation *déterministe*, pour laquelle on cherche à prédire une valeur, et la modélisation *probabiliste*, pour laquelle on cherche à prédire la distribution d'une valeur. Dans ce dernier cas, on parle également de modèle d'*estimation de densité*, conditionnelle ou non selon la présence ou l'absence d'une variable à expliquer.

L'*apprentissage* d'un modèle statistique, aussi appelé inférence ou induction, vise à construire un modèle à partir des données disponibles, ayant des capacités de généralisation sur des données inconnues. Il s'agit d'identifier des relations entre les données valides en général, pas uniquement sur les individus utilisés pour apprendre le modèle. Tout l'enjeu de l'apprentissage statistique est ainsi d'extraire le maximum d'informations à partir des données tout en évitant le phénomène de *sur-apprentissage*.

L'évaluation d'une méthode d'apprentissage statistique est usuellement effectuée suivant le protocole *apprentissage/validation/test*, en partitionnant les données disponibles en trois ensembles (ou échantillons). L'ensemble d'apprentissage est utilisé pour apprendre le modèle. L'ensemble de validation permet d'ajuster les éventuels paramètres de la méthode d'apprentissage. L'ensemble de test, qui contient des individus inconnus pour la méthode d'apprentissage, sert à évaluer les performances en généralisation du modèle appris.

1.1.2 Processus Data Mining

Selon [Fayyad *et al.*, 1996], le Data Mining est un processus non-trivial d'identification de structures inconnues, valides et potentiellement exploitables dans les bases de données. Plusieurs intervenants industriels ont proposé une formalisation de ce processus, sous la forme d'un guide méthodologique nommé CRISP-DM pour CRoss Industry Standard Process for Data Mining [Chapman *et al.*, 2000].

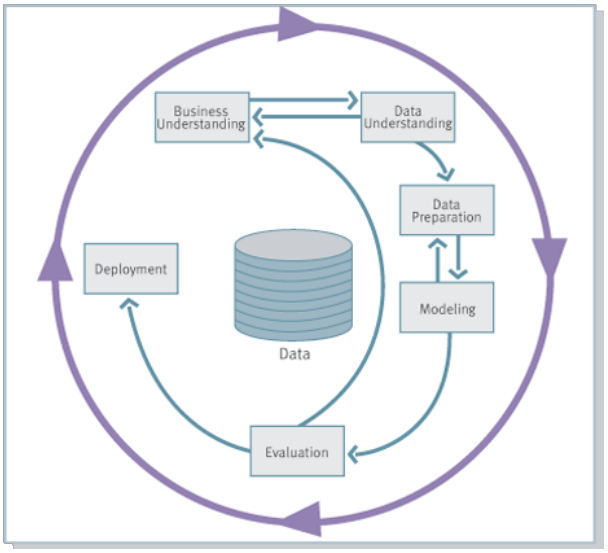


FIG. 1.1 – Processus Data Mining CRISP-DM.

Le modèle CRISP-DM, illustré sur la figure 1.1, propose de découper tout processus Data Mining en 6 phases :

1. La phase de *recueil des besoins* (de l'anglais business understanding), fixe les objectifs industriels et les critères de succès, évalue les ressources, les contraintes et les hypothèses nécessaires à la réalisation des objectifs, traduit les objectifs et critères industriels en objectifs et critères techniques, et décrit un plan de résolution afin d'atteindre les objectifs techniques.
2. La phase de *compréhension des données* (de l'anglais data understanding), réalise la collecte initiale des données, en produit une description, étudie éventuellement quelques hypothèses à l'aide de visualisations et vérifie le niveau de qualité des données.
3. La phase de *préparation des données*, consiste en la construction d'une table de données pour modélisation. Nous nous y intéressons plus particulièrement par la suite.
4. La phase de *modélisation*, procède à la sélection de techniques de modélisation, met en place un protocole de test de la qualité des modèles obtenus, construit les modèles et les évalue selon le protocole de test.

5. La phase d'*évaluation* estime si les objectifs industriels ont été atteints, s'assure que le processus a bien suivi le déroulement escompté et détermine la phase suivante : retour en arrière ou *déploiement*.
6. La phase de *déploiement* industrialise l'utilisation du modèle en situation opérationnelle, définit un plan de contrôle et de maintenance, produit un rapport final et effectue une revue de projet.

Le modèle CRISP-DM est essentiellement un guide méthodologique pour la conduite d'un projet Data Mining. Les phases initiales et finales s'apparentent à des activités d'expertise en organisation, consulting, bases de données et développement informatique. Elles supposent une implication humaine importante. Seules les phases centrales sont partiellement automatisables, notamment, la phase de modélisation, naturellement consommatrice en techniques de modélisation statistique.

1.1.3 Contexte industriel

Le Data Mining à France Télécom couvre une large variété de réalités, dont les principales caractéristiques sont résumées ci-dessous :

- diversité des contextes :
  - . diversité des domaines d'application, comme par exemple le marketing client, le text mining, le web mining, la caractérisation du trafic dans les réseaux, la sociologie ou l'ergonomie,
  - . diversité des tâches, en supervisé ou non supervisé, régression ou classification,
- complexité et volume des données :
  - . diversité des sources d'information, avec des bases de données multiples de plusieurs centaines de tables, des données au format texte, web, image, son,
  - . volume des données, avec des entrepôts de données atteignant la centaine de TeraOctets dans le domaine du marketing client par exemple,
  - . données souvent incomplètes, bruitées, avec des distributions de variables à expliquer très fortement déséquilibrées,
- importance des contraintes :
  - . contraintes fortes en interprétabilité, temps d'apprentissage et temps de déploiement des modèles,
  - . contraintes de réactivité en environnement concurrentiel, limitant de façon drastique les temps d'étude impartis au processus Data Mining.

Exemple d'étude Data Mining à France Télécom. Prenons par exemple le cas d'une étude de scoring client visant à prédire un mois à l'avance les clients résiliant leur abonnement ADSL.

1. Dans la phase de recueil des besoins, on définit les conditions de retour sur investissement en mettant en regard la perte de revenu que représente une résiliation d'abonnement, le coût marketing d'un contact client, par mail, courrier ou téléphone, et le coût informatique de déploiement d'une solution Data Mining.

2. En phase de compréhension des données, les données les plus fines sur les clients, stockées sous la forme d'un entrepôt de données contenant plusieurs centaines de tables relationnelles et des dizaines de TeraOctets, ne sont pas accessibles en temps raisonnable en raison des contraintes informatiques. Les données utilisables directement pour l'étude sont celles d'un datamart marketing, réduit à environ un millier de variables par client. Le taux de résiliation mensuel étant très faible (moins de 1%), un échantillon d'environ 100000 clients est nécessaire pour qualifier de façon significative le phénomène étudié.
3. La phase de préparation des données vise alors à construire de nouvelle variables, sélectionner les variables explicatives pertinentes, et à rechercher une représentation des données informative sous la forme d'un tableau croisé individus * variables.
4. La phase de modélisation exploite une technique de modélisation statistique (réseau de neurones, arbre de décision, régression logistique, classifieur Bayesien naif...) et évalue le taux d'erreur en test selon un protocole apprentissage/validation/test.
5. La phase d'évaluation quantifie l'impact réel d'une action marketing sur les clients ayant la plus forte probabilité de résiliation selon le modèle retenu, lors d'une expérimentation pilote menée en agence.
6. En cas de possibilité de retour sur investissement, le modèle est déployé dans les services marketings opérationnels, ce qui implique un projet de développement informatique à part entière, avec de fortes contraintes sur le nombre de variables explicatives à extraire ou à construire, les temps d'apprentissage et de déploiement des modèles, appliqués potentiellement à des millions de clients.

1.2 Objectif

Après avoir identifié un problème critique pour le processus Data Mining en contexte industriel, nous proposons et précisons une piste de résolution de ce problème, passant par l'automatisation de la phase de préparation des données.

1.2.1 Préparation des données en contexte industriel

La phase de préparation des données est particulièrement importante dans le processus Data Mining [Pyle, 1999, Mamdouh, 2006]. Elle est critique pour la qualité des résultats, et consomme typiquement de l'ordre de 80% du temps d'une étude Data Mining. Dans le cas du Data Mining à France Télécom, le contexte industriel impose des contraintes telles que le potentiel des données collectées dans les systèmes d'information est largement sous-utilisé.

Cette situation s'aggrave années après années, suite à des vitesses d'évolution divergentes des capacités des systèmes d'information, en augmentation très rapide pour le stockage et "seulement" rapide pour le traitement, des capacités de modélisation des méthodes d'apprentissage statistique, en progression lente, et de la disponibilité des analystes de données, au mieux constante. Dans ce contexte, les solutions actuelles sont impuissantes

à répondre à la demande rapidement croissante de l'utilisation de techniques Data Mining dans de nombreux contextes. Les projets, en surnombre, sont abandonnés ou traités sous-optimalement.

Ce goulot d'étranglement peut être résolu en agissant sur deux leviers, par augmentation du nombre d'analystes de données ou par amélioration de l'efficacité du processus Data Mining. Le premier levier est un problème de ressources humaines, le second levier est un problème organisationnel et technique. Nous nous intéressons ici au problème technique de l'automatisation de la phase de préparation des données du processus Data Mining.

1.2.2 Automatisation de la préparation des données

L'automatisation vise à minimiser le niveau d'intervention humaine dans le processus Data Mining. Nous précisons ici dans quel contexte nous recherchons une méthode d'automatisation de la préparation des données.

La phase de préparation des données comprend des étapes de sélection, nettoyage, construction, intégration et recodage des données. Il s'agit de passer d'un ensemble de données dans un format initial provenant potentiellement de plusieurs sources de données à un format préparé de type tableau croisé individus * variables, en entrée de la phase de modélisation. En l'absence d'une description formelle générique des formats initiaux dans leur pluralité, nous faisons l'hypothèse d'un format initial des données, également sous la forme d'un tableau individus * variables.

Les données initiales peuvent provenir de multiples bases de données, être de tout type, bruitées, provenir de variables natives ou construites, sans limitations en nombre d'individus ou de variables. On se donne comme objectif, pour un problème de Data Mining donné, de rechercher une représentation des données performante, c'est à dire un sous-ensemble de variables aboutissant à une modélisation efficace des données en respectant les contraintes du projet Data Mining. Il s'agit donc d'une méthode de sélection de variables pour la préparation des données.

Les méthodes de sélection de variables [Guyon and Elisseeff, 2003, Guyon *et al.*, 2006a] se divisent en approche filtre et approche enveloppe. Dans l'approche filtre, la pertinence des variables est évaluée intrinsèquement par rapport à l'objectif d'un projet Data Mining, alors que dans l'approche enveloppe, il s'agit de rechercher le meilleur sous-ensemble de variables relatif à l'optimisation de la performance d'un modèle statistique. L'approche filtre, générique et généralement efficace en tenue de charge, est adaptée à la phase de préparation des données, alors que l'approche enveloppe, spécifique à un modèle statistique particulier et coûteuse en temps de traitement, est plus adapté à la phase de modélisation. On recherche donc une méthode de sélection de variables selon l'approche filtre.

1.2.3 Critères de qualité d'une méthode de sélection de variables

Nous proposons dans cette section une liste de critères de qualité liés à l'automatisation d'une méthode de sélection de variables en préparation des données.

Généricité. La généricité permet d'appliquer une méthode dans une multitude de contextes,

là où habituellement de multiples méthodes spécialisées sont nécessaires. La généricité s'exprime sur de nombreux axes, parmi lesquels on trouve :
- le domaine applicatif, marketing, texte, web, biologie, écologie, ergonomie...,
- la tâche de modélisation, supervisée ou non supervisée, en classification ou régression, selon le rôle explicatif ou à expliquer des variables,
- la nature de la modélisation, déterministe ou probabiliste,
- le nombre d'individus et de variables, pouvant varier sur plusieurs ordres de grandeurs,
- le type des variables, numériques ou catégorielles, et dans le cas catégoriel le nombre de valeurs, pouvant varier comme le nombre d'individus sur plusieurs ordres de grandeurs,
- la densité des variables, c'est à dire la proportion des valeurs non nulles ou non vides,
- le déséquilibre ou non de la distribution des variables, notamment pour la variable à expliquer dans le cas de la classification supervisée,
- l'indépendance ou la corrélation entre les variables,
- la présence de bruit dans les données,
- ...

Absence de paramétrage. L'absence de paramétrage est un élément critique pour l'automatisation des méthodes de préparation des données, puisqu'elle permet d'éviter toute phase d'ajustement des paramètres, coûteuse en temps et génératrice d'incertitude.

Fiabilité. La fiabilité, aussi appelée robustesse, représente la capacité d'une méthode à extraire des informations valides en général, pas uniquement sur les individus considérés en apprentissage. Il s'agit en fait de contrôler au mieux le phénomène du sur-apprentissage.

Finesse. La finesse représente la capacité à détecter des informations (ou motifs, structures, patterns) avec le moins d'individus possible.

Fiabilité et finesse sont deux critères fondamentaux pour toute méthode de modélisation statistique. Face au coût de déploiement des modèles, il est essentiel de rechercher la fiabilité des informations détectées. De façon concurrente, les gains potentiels d'un déploiement peuvent être tels que la moindre information valide est recherchée activement. Sous la pression de ces enjeux antagonistes, les critères de fiabilité et de finesse sont très importants pour l'analyste de données, en étant générateur de confiance, donc de diminution du temps d'analyse, de test et de vérification.

Interprétabilité. L'interprétabilité permet d'une part de vérifier le déroulement correct du processus Data Mining, et d'autre part de faciliter l'exploitation des résultats d'analyse, notamment dans le cas d'une étude à but descriptif ou exploratoire.

Efficacité. La tenue de charge s'exprime sur plusieurs axes :
- par rapport au nombre d'individus et de variables,
- en modélisation et en déploiement des modèles,
- pour les accès aux bases de données, en occupation mémoire, et en temps de traitement.

1.3 Contribution

Dans ce mémoire, nous présentons une méthode de sélection de variables en préparation des données baptisée MODL.

La méthode MODL repose sur l'introduction d'une famille de modèles non paramétriques pour l'estimation de densité : les *modèles en grille de données*. Chaque variable étant découpée en intervalles ou groupes de valeurs selon sa nature numérique ou catégorielle, l'espace complet des données est partitionné en une grille de cellules résultant du produit cartésien de ces partitions univariées. On recherche alors un modèle où l'estimation de densité est constante sur chaque cellule de la grille. Cette famille de modèle s'applique à tous les cas de figure d'un tableau croisé individus * variables, quel que soit le type, numérique ou catégorielle, ou le rôle, explicatif ou à expliquer, des variables. Les modèles envisagés, qui ne font pas d'hypothèse sur la distribution des données ou sur le domaine applicatif sont des approximateurs universels, en ce sens qu'il peuvent asymptotiquement approximer n'importe quelle distribution de densité conditionnelle ou jointe.

Les modèles en grille de données s'apparentent dans le cas univarié supervisé à la discrétisation supervisée ou au groupement de valeurs supervisé utilisés classiquement dans les arbres de décision [Zighed and Rakotomalala, 2000]. Dans le cas général, les modèles en grille se rapprochent des méthodes de discrétisation multivariée ou des tables de décision [Kohavi, 1995]. L'originalité de la méthode MODL se situe non pas dans la famille de modèles envisagée, mais dans la démarche utilisée permettant de rechercher le meilleur modèle prédictif.

La démarche de modélisation emprunte à la fois à l'approche Bayesienne et au principe MDL (minimum description length) [Rissanen, 1978], qui sont deux techniques classiques de sélection de modèles. De façon plus atypique, l'espace des modèles dépend des données, ce qui permet de réutiliser au maximum les informations considérées comme acquises, notamment les informations de nature explicative, hors du champ de modélisation. L'utilisation de modèles dépendant des données impose néanmoins des précautions particulières pour éviter le sur-apprentissage. L'application de cette démarche de modélisation aboutit pour les modèles en grille de données à un critère d'évaluation analytique résultant d'un calcul exact, et non d'une approximation, ce qui les rend utilisables y compris dans un cadre non asymptotique. Des algorithmes d'optimisation combinatoire nouveaux sont introduits pour optimiser efficacement ce critère.

De façon synthétique, notre contribution est l'introduction d'une nouvelle méthode de sélection de variables en préparation des données, basée sur des modèles en grille de données pour l'estimation non paramétrique de densité. Cette méthode se caractérise par une démarche innovante de modélisation dépendant des données et par de nouveaux algorithmes d'optimisation combinatoire performants.

1.4 Organisation du document

Dans le chapitre 2, nous introduisons les modèles en grille de données et leur critère d'évaluation de façon didactique, en partant du cas le plus simple, celui de la classification supervisée dans le cas d'une seule variable explicative numérique pour une seule variable à expliquer catégorielle. Ce cas est étendu progressivement en prenant en compte tout type de variable, catégoriel ou numérique, tout rôle de variable, explicatif ou à expliquer, et tout nombre de variables, de l'univarié au multivarié.

Dans le chapitre 3, nous résumons de façon synthétique les algorithmes introduits pour optimiser les modèles en grille de données. Les cas univariés de la discrétisation supervisée et du groupement de valeurs supervisé sont abordés en premier, avant de passer au cas multivarié le plus général. Les détails des algorithmes sont fournis en annexe A pour la discrétisation, en annexe B pour le groupement de valeurs, et en annexe D dans le cas particulier des grilles bivariées.

Dans le chapitre 4, nous présentons une évaluation approfondie des modèles en grille de données. L'évaluation est d'abord théorique, en étudiant les seuils de détection d'information dans la limite des échantillons de petite taille. L'évaluation se poursuit de façon expérimentale au moyen de jeux de données réels et synthétiques, de façon à valider l'utilisation des modèles en grille dans la phase de préparation des données et dans la phase de modélisation, dans des contextes supervisés et non supervisés. La méthode est ensuite confrontée aux meilleures méthodes de l'état de l'art lors de challenges internationaux. Enfin, son utilisation est présentée dans le contexte industriel du Data Mining à France Télécom. Les annexes A, B et C fournissent des évaluations détaillées complémentaires dans les cas univariés et bivariés.

Dans le chapitre 5, nous dressons un panorama comparatif des méthodes de l'état de l'art en discrétisation et groupement de valeurs univarié supervisé, puis en discrétisation multivariée. Nous récapitulons ensuite notre démarche de modélisation, dépendante des données, en la positionnant par rapport aux démarches fondées sur l'approche Bayesienne et sur le principe de description de longueur minimum.

Enfin, dans le chapitre 6, nous dressons un bilan de la méthode, en précisant son état de maturité, ses limites, et ce qui reste à effectuer, notamment en matière d'évaluation. Nous proposons à titre de perspectives plusieurs pistes pour étendre l'automatisation de la phase de préparation des données du processus Data Mining, en amont par la prise en compte de formats de données multi-tables et de la connaissance métier quand elle est disponible (comme illustré en annexe E), en aval par le développement de nouvelles méthodes de modélisation exploitant le potentiel des modèles en grille de données.

Le parti pris d'aborder l'état de l'art après la présentation de notre méthode est lié à sa généricité, puisqu'elle traite de problèmes de préparation des données, de régression, de classification supervisée et non supervisée et de sélection de modèles. Cette diversité de sujets rend vaine toute tentative préliminaire d'état de l'art exhaustif et pertinent. Nous avons donc choisi délibérément de décrire la méthode au préalable dans le chapitre 2, en expliquant au fur et à mesure les différents contextes d'application. Dans le chapitre 5, la méthode est positionnée par rapport à l'état de l'art en sélectionnant les aspects les plus pertinents.

2

Modèles en grille de données

Sommaire

On introduit dans ce chapitre une famille de modèles non paramétriques pour l'estimation de densité. Chaque variable étant découpée en intervalles ou groupes de valeurs selon sa nature numérique ou catégorielle, l'espace complet des données est partitionné

en une grille de cellules résultant du produit cartésien de ces partitions univariées. On recherche alors un modèle où l'estimation de densité est constante sur chaque cellule de la grille. Afin de trouver la meilleure estimation de densité connaissant les données, on applique une approche Bayesienne de la sélection de modèles. En proposant une distribution a priori des modèles exploitant la hiérarchie des paramètres de modélisation, on aboutit à un critère d'évaluation permettant d'évaluer exactement la probabilité qu'un modèle explique les données observées. Cette démarche est appliquée à tous les cas de figure : apprentissage supervisé (classification et régression) et non supervisé.

Dans la suite, on se focalisera sur les échantillons de données contenant un nombre fini d'individus et de variables, les variables étant soit catégorielles, soit numériques. Il s'agit de la représentation classique des données sous forme de tableau individus * variables.

La section 2.1 traite du problème de la classification supervisée pour une variable à expliquer catégorielle, en abordant successivement le cas de la discrétisation d'une variable explicative numérique, du groupement des valeurs d'une variable catégorielle, puis en généralisant l'approche au cas bivarié et enfin au multivarié qui incorpore explicitement un aspect sélection de variables. Les modèles en grille présentés sont des estimateurs probabilistes de la variable catégorielle à expliquer conditionnellement aux variables explicatives. La section 2.2 étend ces modèles au cas de la régression d'une variable à expliquer numérique en proposant de discrétiser également la variable à expliquer. La section 2.3 généralise la formalisation usuelle de la classification supervisée, qui suppose implicitement un faible nombre de valeurs catégorielles à expliquer. Cette généralisation est effectuée au moyen d'un groupement des valeurs de la variable à expliquer. La section 2.4 traite du cas non supervisé en utilisant les modèles en grille pour la description des corrélations entre les variables, ce qui peut s'interpréter comme un modèle d'estimation de densité jointe. La section 2.5 généralise l'utilisation des modèles en grille pour l'estimation de densité d'un nombre quelconque de variables à expliquer conditionnellement aux variables explicatives. Enfin, la section 2.6 synthétise ce chapitre.

L'objectif de ce chapitre est d'introduire la famille de modèles en grille de façon progressive et didactique, en partant des objectifs d'apprentissage visés et en aboutissant à la définition des modèles et à un critère permettant d'évaluer leur pertinence. Du fait de ce parti pris didactique, ce chapitre est assez volumineux. On pourra se limiter dans un premier temps à la section 2.1 relative à la classification supervisée pour une compréhension de l'approche, les autres sections ayant le rôle de formulaire de référence pour l'application de l'approche dans tous les cas de figure.

Les critères d'évaluation des modèles en grille sont introduits délibérément suivant une démarche Bayesienne de la sélection de modèles, essentiellement pour des raisons de simplification de la présentation. On reviendra sur ce positionnement de l'approche dans le chapitre 5.

Les algorithmes d'optimisation des modèles en grille sont présentés dans le chapitre 3, et leur évaluation sur des jeux de données artificiels et réels dans le chapitre 4.

2.1 Classification supervisée

Dans un problème de classification supervisée, l'échantillon de données est constitué de N individus, K variables explicatives $X_1, X_2, \ldots, X_K$ et d'une variable à expliquer catégorielle Y ayant J valeurs. Dans le cas déterministe, on cherche à modéliser la variable à expliquer comme une fonction des variables explicatives, sous la forme $Y = f(X_1, X_2, \ldots, X_K)$. Dans un cadre plus général, on cherche à estimer la dépendance entre la variable à expliquer et les variables explicatives sous une forme probabiliste $P(Y|X_1, X_2, \ldots, X_K)$.

Le problème étant complexe en général, on procède usuellement dans une première phase de statistique descriptive à une analyse univariée, où les variables explicatives sont examinées une à une indépendamment. L'objectif principal de l'analyse univariée est de déterminer si une variable explicative est corrélée avec la variable à expliquer, autrement dit si elle contient de l'information permettant de prédire la valeur de la variable à expliquer. L'analyse univariée permet ainsi en pratique de détecter les variables explicatives informatives, de comparer deux variables explicatives selon leur importance prédictive et d'éliminer les variables non corrélées avec la variable à expliquer.

Dans cette section, on introduit les modèles d'estimation de densité conditionnelle dans le cas univarié au moyen de modèles de discrétisation pour les variables numériques et de groupement de valeurs pour les variables catégorielles. L'approche est ensuite étendue au cas bivarié, puis au cas multivarié pour un nombre quelconque de variables explicatives.

2.1.1 Discrétisation

La variable explicative considérée est ici numérique. Après avoir introduit le problème au moyen d'un exemple illustratif, on propose un modèle d'estimation de densité conditionnelle basé sur la discrétisation de la variable explicative puis un critère d'évaluation de la qualité d'un modèle fondé sur une approche Bayesienne.

2.1.1.1 Présentation

La base Iris [Fisher, 1936] comporte 150 individus. Chaque individu représente une fleur de la famille des iris. L'objectif est ici de caractériser la variété d'iris (Versicolor, Virginica ou Setosa) d'un individu connaissant la largeur et la longueur de ses pétales et sépales. Chaque individu est donc décrit par quatre variables explicatives, toutes numériques. On s'intéresse ici à la variable largeur de sépale, en essayant de déterminer son degré de corrélation avec la variété d'iris. La figure 2.1 reporte pour chaque valeur de largeur de sépale le nombre d'individus de l'échantillon par variété d'iris. Par exemple, parmi les 26 iris dont la largeur de sépale est égale à 3.0 cm, 12 sont de la variété Virginica, 8 de la variété Versicolor et 6 de la variété Setosa.

La question que l'on se pose est : que peut-on dire de la variété d'un iris connaissant la largeur de ses sépales ? En se basant sur la figure 2.1, une attitude prudente consiste à décréter que la variable largeur de sépale n'est pas informative, les pics de la figure étant probablement dus au hasard. A l'inverse, on pourrait décider que chaque pic est significatif et que, par exemple, la variété Virginica est la plus fréquente pour les largeurs

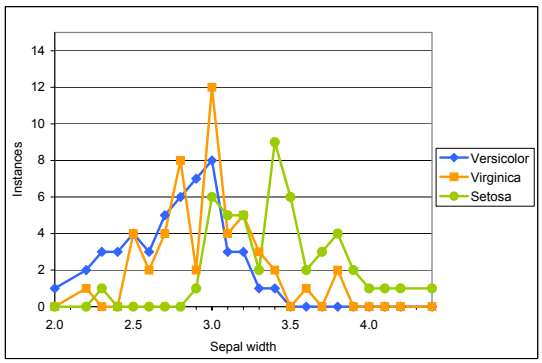


FIG. 2.1 – Nombre d'individus par variété d'Iris pour chaque valeur de largeur de sépale. La base est globalement équidistribuée sur les variétés d'Iris.

de sépale 2.8 et 3.0, mais qu'elle est dépassée par la variété Versicolor pour la largeur 2.9. Cela paraît néanmoins risqué, étant donné le faible nombre d'individus de l'échantillon. Il s'agit donc de trouver et de quantifier un compromis acceptable entre des prédictions précises, mais potentiellement risquées, et des prédictions plus générales mais robustes.

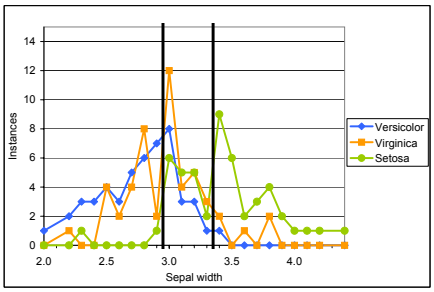
On propose d'estimer l'information prédictive contenue dans la variable explicative au moyen d'un modèle de discrétisation. Autrement dit, on partitionne les largeurs de sépale en intervalles, ce qui permet d'estimer par comptage la proportion de chaque variété d'iris sur chaque intervalle. Cette famille de modèles est suffisamment riche pour exprimer toutes les stratégies de prédiction, prudentes ou risquées. Si par exemple on ne retient qu'un seul intervalle, on ne pourra prédire que la répartition moyenne $(1/3, 1/3, 1/3)$ des variétés d'iris sur l'échantillon, ce qui constitue une prédiction robuste mais peu informative. On peut à l'inverse créer autant d'intervalles qu'il y a de valeurs de largeur de sépale, ce qui correspond à une explication très précise mais intuitivement trop risquée dans un but prédictif. La figure 2.2 propose un compromis plus raisonnable où la variable largeur de sépale est découpée en se basant sur une discrétisation en trois intervalles. Dans le premier intervalle (largeur de sépale inférieure à 2.95 cm), la variété Setosa est minoritaire, dominée essentiellement par la variété Versisolor et dans une moindre mesure par la variété Virginica. Dans le deuxième intervalle, les trois variétés d'iris sont représentées également. Dans le troisième intervalle (largeur de sépale supérieure à 3.35 cm), la variété Setosa est largement majoritaire.

Problèmes de sélection de modèles. Les modèles de discrétisation sont particulièrement expressifs, mais se pose le problème du choix du meilleur modèle :

- Quel est le bon nombre d'intervalles ?
- Comment choisir les bornes des intervalles ?
- Comment caractériser l'absence d'information discriminante ?

En sus du problème du choix du bon modèle, on peut également se poser les questions suivantes :

- Comment comparer deux discrétisations d'une même variable explicative ?
- Comment comparer l'importance prédictive de deux variables explicatives distinctes ?



Total	57	57	36
Versicolor	34	15	1
Virginica	21	24	5
Setosa	2	18	30
	$]-\infty, 2.95[$	$[2.95, 3.35[$	$[3.35, +\infty[$

FIG. 2.2 – Nombre d'individus par variété d'Iris pour une discrétisation en trois intervalles de la variable largeur de sépale.

2.1.1.2 Formalisation

Soit X une variable explicative numérique, Y une variable à expliquer catégorielle avec J valeurs et $D = \{(x_n, y_n); 1 \leq n \leq N\}$ un échantillon de données de (X, Y).

On recherche un modèle d'estimation de la densité conditionnelle $P(Y|X)$ de Y sachant X.

Les choix 2.1 et 2.2 sont à la base de l'approche présentée dans ce chapitre. Ils seront analysés dans le chapitre 5.

Choix 2.1. Choix de la statistique d'ordre.
En considérant qu'une bonne estimation de densité conditionnelle $P(Y|X)$ devrait être invariante par toute transformation monotone de la variable explicative et peu sensible aux valeurs atypiques (outliers), on se base sur la statistique d'ordre. Il s'agit alors de prédire la valeur de Y connaissant le rang de X, et non pas connaissant la valeur de X.

Choix 2.2. Choix de la précision des modèles.
On dispose d'un échantillon de données de taille finie N, ce qui rend peu raisonnable la possibilité d'approximer la véritable densité conditionnelle avec une précision meilleure que $1/N$. On va donc se restreindre à des modèles limités dans leur expressivité à un nombre fini de fréquences (plutôt que des probabilités définies continûment sur $[0, 1]$), en se basant sur le comptage des individus dans l'échantillon.

La définition 2.1 tire partie de ces choix pour décrire une famille de modèles de discrétisation, pour laquelle la densité conditionnelle $P(Y|X)$ est constante par intervalle. Par facilité de langage, on parlera de modèle de discrétisation plutôt que de modèle d'estimation de densité conditionnelle.

Définition 2.1. Un modèle de discrétisation standard est défini par :
- un nombre d'intervalles,
- une partition de la variable explicative en intervalles, spécifiée sur les rangs des valeurs explicatives,
- la distribution des valeurs de la variable à expliquer par intervalle, spécifiée par les effectifs de chaque valeur à expliquer localement à l'intervalle.

On dira qu'un tel modèle de discrétisation est de type SDM (Standard Discretization Model).

Notations 2.1.
- N : nombre d'individus de l'échantillon
- J : nombre de valeurs de la variable à expliquer (connu)
- I : nombre d'intervalles (inconnu)
- $N_{i.}$: nombre d'individus de l'intervalle i
- N_{ij} : nombre d'individus de l'intervalle i pour la valeur à expliquer j

Un modèle de discrétisation standard est entièrement caractérisé par le choix des paramètres $I, \{N_{i.}\}_{1\leq i\leq I}, \{N_{ij}\}_{1\leq i\leq I, 1\leq j\leq J}$.

Il s'agit maintenant de sélectionner le meilleur modèle M en se basant sur les données disponibles de l'échantillon D, à savoir le modèle le plus probable connaissant les données. En appliquant l'approche Bayesienne, cela revient à maximiser

$$P(M|D) = \frac{P(M)P(D|M)}{P(D)}. \tag{2.1}$$

La probabilité des données $P(D)$ étant constante dans ce problème de maximisation, on se ramène à la maximisation de $P(M)P(D|M)$. En se basant sur la structure naturelle des paramètres d'un modèle de discrétisation M, on obtient

$$P(M)P(D|M) = P(I)P\big(\{N_{i.}\}|I\big)P\big(\{N_{ij}\}|I, \{N_{i.}\}\big)P(D|M). \tag{2.2}$$

On ajoute ici une hypothèse, en recherchant les modèles tels que les distributions des valeurs de la variable à expliquer soient indépendantes par intervalle. Cette hypothèse possède d'une part un intérêt explicatif en se focalisant sur les modèles des comportements discriminants par intervalle, d'autre part un intérêt mathématique en permettant l'évaluation des modèles par une formule analytique, et enfin un intérêt informatique, en facilitant la mise au point d'algorithmes d'optimisation efficaces (voir chapitre 3). Il est à noter que l'indépendance par intervalle est sous-tendue par l'hypothèse classique de distribution indépendante par individu (hypothèse IID).

En notant D_i la sous partie de D restreinte à l'intervalle i, on obtient

$$P(M)P(D|M) = P(I)P\big(\{N_{i.}\}|I\big)\prod_{i=1}^{I} P\big(\{N_{ij}\}|I, \{N_{i.}\}\big)\prod_{i=1}^{I} P(D_i|M). \tag{2.3}$$

Pour finaliser l'évaluation d'un modèle, il faut proposer une distribution a priori des paramètres d'un modèle de discrétisation. La définition 2.2 formalise un tel choix, en explicitant l'hypothèse d'indépendance par intervalle et en proposant une distribution uniforme des paramètres à chaque étage de la hiérarchie des paramètres des modèles.

Définition 2.2. On appelle a priori hiérarchique l'a priori de modèle SDM basé sur les hypothèses suivantes :
- le nombre d'intervalles est compris entre 1 et N de façon équiprobable,
- pour un nombre d'intervalles donné, toutes les partitions en intervalles des rangs de la variable explicative sont équiprobables,
- pour un intervalle donné, toutes les distributions des valeurs de la variable à expliquer sont équiprobables,

– les distributions des valeurs de la variable à expliquer sur chaque intervalle sont indépendantes les unes des autres.

En utilisant la définition formelle des modèles SDM et de leur distribution a priori hiérarchique, la formule de Bayes permet de calculer de manière exacte la probabilité d'un modèle connaissant les données, ce qui conduit au théorème 2.1.

Théorème 2.1. *Un modèle de discrétisation standard suivant un a priori hiérarchique est optimal au sens de Bayes si son évaluation par la formule suivante est minimale sur l'ensemble de tous les modèles :*

$$\log N + \log \binom{N+I-1}{I-1} + \sum_{i=1}^{I} \log \binom{N_{i.}+J-1}{J-1} + \sum_{i=1}^{I} \log \frac{N_{i.}!}{N_{i1}!N_{i2}!\dots N_{iJ}!} \qquad (2.4)$$

La notation $\binom{n}{k}$ représente le coefficient binomial.

Démonstration. La probabilité a priori d'un modèle de discrétisation M est définie comme la probabilité a priori des paramètres du modèle $\{I, \{N_{i.}\}_{1\leq i\leq I}, \{N_{ij}\}_{1\leq i\leq I, 1\leq j\leq J}\}$.

En utilisant la hiérarchie du paramétrage et l'hypothèse d'indépendance des distributions entre les intervalles, nous avons établi précédemment la formule 2.3

$$P(M)P(D|M) = P(I)P\big(\{N_{i.}\}|I\big)\prod_{i=1}^{I} P\big(\{N_{ij}\}|I,\{N_{i.}\}\big)\prod_{i=1}^{I} P(D_i|M)$$

qui représente la probabilité a posteriori d'un modèle de discrétisation M, au terme $P(D)$ près.

Précisons maintenant la valeur de chacun des termes de cette formule, en utilisant les hypothèses de l'a priori hiérarchique introduit en définition 2.2.

La première hypothèse de l'a priori est que le nombre d'intervalles est compris entre 1 et N de façon équiprobable. On obtient dès lors

$$P(I) = \frac{1}{N}.$$

La deuxième hypothèse de l'a priori est que le nombre d'intervalle I étant fixé, toutes les discrétisations en I intervalles sont équiprobables. Le calcul de la probabilité a priori d'une discrétisation particulière se ramène au problème du dénombrement du nombre de façons de diviser une séquence de N individus en I sous-séquences représentant les intervalles. Cela est équivalent au nombre de façons de décomposer un nombre N comme une somme de I entiers représentant les effectifs de chaque intervalle. Ce dénombrement combinatoire fournit $\binom{N+I-1}{I-1}$ possibilités pour le choix des effectifs par intervalles $\{N_{i.}\}_{1\leq i\leq I}$. Ces paramètres de discrétisation étant équiprobables, on obtient

$$P\big(\{N_{i.}\}|I\big) = \frac{1}{\binom{N+I-1}{I-1}}.$$

La troisième hypothèse de l'a priori indique que pour un intervalle i donné d'effectif $N_{i.}$, toutes les distributions des J valeurs sont équiprobables. Il s'agit ici de spécifier les

paramètres d'une distribution multinômiale de $N_{i.}$ individus sur J valeurs. Là encore, on se ramène à un problème de dénombrement, celui du nombre de façons de décomposer un nombre $N_{i.}$ comme une somme de J termes. Ces paramètres de distribution multinômiale étant supposés équiprobables, on obtient

$$P\big(\{N_{ij}\}|I,\{N_{i.}\}\big) = \frac{1}{\binom{N_{i.}+J-1}{J-1}}.$$

Les termes d'a priori étant précisés, il nous reste à évaluer le terme de vraisemblance pour chaque intervalle, c'est à dire à calculer la probabilité d'observer les données d'un intervalle i d'effectif $N_{i.}$ connaissant le modèle de distribution multinômiale défini sur l'intervalle. Le nombre de façons d'observer $N_{i.}$ individus distribués suivant les paramètres d'une multinômiale est donné par la formule du multinôme $\frac{N_{i.}!}{N_{i1}!N_{i2}!...N_{iJ}!}$. Toutes ces observations étant supposées équiprobables, la vraisemblance par intervalle est égale à

$$\frac{1}{\frac{N_{i.}!}{N_{i1}!N_{i2}!...N_{iJ}!}}.$$

En remplaçant dans la formule 2.3 chaque terme d'a priori et de vraisemblance, on obtient la formule

$$P(M)P(D|M) = \frac{1}{N}\frac{1}{\binom{N+I-1}{I-1}}\prod_{i=1}^{I}\frac{1}{\binom{N_{i.}+J-1}{J-1}}\prod_{i=1}^{I}\frac{1}{\frac{N_{i.}!}{N_{i1}!N_{i2}!...N_{iJ}!}}.$$

Par passage au log négatif, on établit la formule 2.4 du théorème 2.1. □

Interprétation synthétique. La formule 2.4 est l'opposé d'une log-vraisemblance. Cette utilisation du log négatif des probabilités correspond à une quantité d'information [Shannon, 1948]. Le premier terme correspond au choix du nombre d'intervalles et le second terme au choix des bornes des intervalles. Le troisième terme représente le choix des distributions de la variable à expliquer dans chaque intervalle et le dernier terme la probabilité d'observer les valeurs de la variable à expliquer connaissant le modèle de discrétisation.

Remarque 2.1. *La démonstration du théorème 2.1 montre comment l'utilisation de la définition 2.2 de l'a priori sur les paramètres d'une discrétisation permet de ramener le calcul des probabilités a priori et a posteriori à des problèmes de dénombrement. A chaque niveau de décision, les choix étant supposés uniformes, il suffit de les dénombrer. Cette méthode est à la base de tous les critères d'évaluation des modèles en grille. Dans la suite de ce chapitre, les paramètres des modèles et les distributions a priori sur ces paramètres sont précisément définis, ce qui permet de reproduire si nécessaire les preuves des théorèmes (non fournies dans le reste du document).*

Remarque 2.2. *La démonstration du théorème 2.1 permet d'obtenir un critère analytique pour l'évaluation exacte de la log-vraisemblance d'un modèle de discrétisation. Les hypothèses utilisées pour aboutir à ce critère, en particulier l'utilisation des rangs plutôt que des valeurs et l'utilisation de distributions de probabilité discrètes plutôt que continues, seront discutées dans le chapitre 5.*

Réponses aux problèmes de sélection de modèles. On peut maintenant répondre aux questions posées en fin de section 2.1.1.1. Le nombre d'intervalles ainsi que leurs bornes sont déterminés par minimisation du critère 2.4. L'absence d'information discriminante se caractérise par une discrétisation de la variable explicative en un seul intervalle. Dans ce cas, la formule 2.4 se réduit à

$$\log N + \log \binom{N+J-1}{J-1} + \log \frac{N!}{N_1!N_2!\ldots N_J!}, \tag{2.5}$$

en notant N_j l'effectif de la valeur à expliquer j. Cela correspond à la probabilité a posteriori d'un modèle multinômial des valeurs de la variable à expliquer, indépendamment de la variable explicative.

Pour comparer deux discrétisations M et M' d'une même variable explicative, on peut utiliser l'interprétation probabiliste du coût $c(M) = -\log P(M) - \log P(D|M)$ de la formule 2.4. On en déduit que le ratio entre les probabilité a posteriori des modèles M et M' s'exprime en fonction de la différence des coûts $c(M)$ et $c(M')$ selon :

$$\log \frac{P(M|D)}{P(M'|D)} = -(c(M) - c(M')). \tag{2.6}$$

Cette relation est également utilisable pour comparer l'importance prédictive de deux variables explicatives distinctes. En effet, le coût $c(M_\star)$ de la discrétisation optimale d'une variable explicative X peut s'interpréter comme la probabilité que la variable X explique Y sur la base d'un modèle de discrétisation.

A titre d'exemple, la discrétisation optimale $M_\star$ de la variable largeur de sépale de la base Iris est présentée sur la figure 2.2. Le coût de cette discrétisation en trois intervalles est $c(M_\star) = 148.18$. La discrétisation $M_\emptyset$ (modèle nul) en un seul intervalle a un coût $c(M_\emptyset) = 165.86$. En utilisant la formule 2.6, l'hypothèse d'une information prédictive basée sur les trois intervalles du modèle optimal est de l'ordre de cinquante millions de fois plus probable que l'hypothèse d'indépendance de la variable explicative du modèle nul ($e^{-(c(M_\star)-c(M_\emptyset))} = 47407904.5$).

2.1.2 Groupement de valeurs

On considère maintenant le cas de la classification supervisée avec une variable explicative catégorielle. On propose dans cette section un modèle d'estimation de densité conditionnelle basé sur le groupement des valeurs de la variable explicative.

2.1.2.1 Présentation

La base Mushroom de l'UCI [Blake and Merz, 1996] est une base dont les individus sont des champignons décrits par une vingtaine de variables explicatives catégorielles, telles la forme, la taille, la couleur du chapeau, du pied ou des spores. La variable à expliquer est la classe du champignon : comestible (edible) ou poison (poisonous). On s'intéresse ici à la variable couleur de chapeau en essayant de déterminer son degré de corrélation avec la classe du champignon. On présente dans le tableau 2.1 les 10 valeurs de la variable

TAB. 2.1 – Effectif et proportion de champignons comestibles ou poisons par couleur de chapeau.

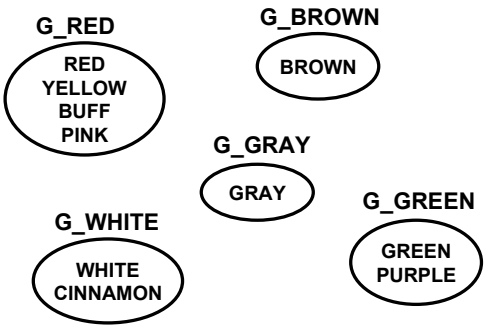
Valeur	edible	poisonous	Effectif
BROWN	55.2%	44.8%	1610
GRAY	61.2%	38.8%	1458
RED	40.2%	59.8%	1066
YELLOW	38.4%	61.6%	743
WHITE	69.9%	30.1%	711
BUFF	30.3%	69.7%	122
PINK	39.6%	60.4%	101
CINNAMON	71.0%	29.0%	31
GREEN	100.0%	0.0%	13
PURPLE	100.0%	0.0%	10

explicative couleur de chapeau, triées par effectif décroissant, en reportant les proportions par classe de champignon.

Les informations portées par le tableau 2.1 paraissent suffisantes pour évaluer la corrélation entre couleur de chapeau et classe de champignon. Néanmoins, la question de la fiabilité d'une prédiction basée sur la couleur du chapeau n'est pas résolue. Par exemple, est-il raisonnable de manger un champignon au chapeau vert en se basant sur une expérience limité à une dizaine de champignons ? De même, il paraît intéressant de synthétiser les connaissances apprises en regroupant les catégories de champignons présentant des caractéristiques similaires vis-à-vis de leur classe. Ainsi, les champignons dont le chapeau est rouge ou jaune ont en commun une proportion de comestible environ égale à 40%. Il paraît légitime de les grouper. Inversement, les champignons au chapeau rouge ou jaune semblent clairement différenciés des champignons au chapeau blanc (70% de comestibles).

On propose d'estimer l'information prédictive contenue dans la variable explicative au moyen d'un modèle de groupement de valeurs. Autrement dit, on partitionne les couleurs de chapeau en groupes de couleurs, ce qui permet d'estimer par comptage la proportion de chaque classe de champignon par groupe de couleurs. Cette famille de modèles permet d'exprimer toutes les stratégies de prédiction, en passant de la plus prudente (un seul groupe signifie qu'il n'y a pas d'information prédictive discriminante) à la plus précise (autant de groupes que de valeurs). La figure 2.3 propose un groupement des couleurs de chapeau réalisant un compromis entre discrimination et robustesse. Les couleurs de chapeau gris ou marron sont isolées dans deux groupes séparés parce que leur (faible) différence de discrimination de la classe de champignon est significative en raison des effectifs importants concernés. A l'inverse, la couleur chamois (buff) est regroupée avec les couleurs rouge, jaune et rose en dépit d'une différence de 10 points dans le taux de champignons comestibles : le nombre de champignons au chapeau chamois n'est pas suffisant pour que la différence soit significative.

L'enjeu d'un bon groupement de valeurs est de trouver un bon compromis entre discrimination et fiabilité. Quand le nombre de valeurs de la variable explicative devient



Groupe	edible	poisonous	Effectif
G_RED	38.9%	61.1%	2032
G_BROWN	55.2%	44.8%	1610
G_GRAY	61.2%	38.8%	1458
G_WHITE	69.9%	30.1%	742
G_GREEN	100.0%	0.0%	23

FIG. 2.3 – Exemple de groupement des couleurs de chapeau de champignon.

important, la recherche d'un bon groupement devient plus difficile en raison du risque accru de sur-apprentissage. Dans la situation extrême où le nombre de valeurs explicatives est égal au nombre d'individus, toute inférence correspond à un apprentissage par coeur des individus, qui ne pourra se généraliser à des individus non déjà rencontrés. Dans ce cas, la meilleure décision de groupement est de constituer un seul groupe. Dans les applications réelles du Data Mining, certains domaines requièrent un groupement des valeurs catégorielles. Dans le cas des applications marketing par exemple, des variables telles que le pays, la ville, le code postal, le prénom, le code produit, ont habituellement de très nombreuses valeurs. Il est alors nécessaire de prétraiter ces variables par des regroupements de valeurs pour produire des modèles de classification performants.

De façon similaire à la discrétisation, les modèles de groupement sont très expressifs, et le problème du choix du meilleur modèle se pose sous la forme :
- Quel est le bon nombre de groupes ?
- Comment choisir la partition des valeurs explicatives en groupes ?
- Comment caractériser l'absence d'information discriminante ?

2.1.2.2 Formalisation

Soit X une variable explicative catégorielle, Y une variable à expliquer catégorielle avec J valeurs et $D = \{(x_n, y_n); 1 \leq n \leq N\}$ un échantillon de données (X, Y).

On recherche un modèle d'estimation de la densité conditionnelle $P(Y|X)$ de Y sachant X.

On dispose d'un échantillon de données de taille finie N, ce qui rend peu raisonnable la possibilité d'approximer la véritable densité conditionnelle avec une précision meilleure que $1/N$. On va donc limiter l'expressivité des modèles aux estimations de probabilité par fréquence, comme dans le choix 2.2 introduit pour la discrétisation supervisée.

On propose dans la définition 2.3 une famille de modèles de groupement de valeurs, pour laquelle la densité conditionnelle $P(Y|X)$ est constante par groupe. Par facilité de langage, on parlera de modèle de groupement plutôt que de modèle d'estimation de densité conditionnelle.

Définition 2.3. Un modèle de groupement standard est défini par :

- un nombre de groupes,
- une partition de la variable explicative en groupes de valeurs,
- la distribution des valeurs de la variable à expliquer par groupe, spécifiée par les effectifs de chaque valeur à expliquer localement au groupe.

On dira qu'un tel modèle de groupement est de type SGM (Standard Grouping Model).

Notations 2.2.
- N : nombre d'individus de l'échantillon
- J : nombre de valeurs de la variable à expliquer (connu)
- V : nombre de valeurs de la variable explicative
- I : nombre de groupes de la variable explicative (inconnu)
- $i(v)$: index du groupe auquel est rattaché la valeur explicative v
- $N_{i.}$: nombre d'individus du groupe i
- N_{ij} : nombre d'individus du groupe i pour la valeur à expliquer j

La variable X étant connue, le nombre de ses valeurs ainsi que leur effectif dans l'échantillon D font partie des connaissances initiales utilisables pour la modélisation de la densité conditionnelle $P(Y|X)$. La partition des valeurs en groupes fait quant à elle partie du paramétrage du modèle de groupement. Pour une partition donnée, les effectifs par groupe $N_{i.}$ sont déduits des effectifs par valeur explicative. Par contre, la distribution des valeurs à expliquer sur chaque groupe fait partie des paramètres de modélisation. En résumé, tout ce qui concerne la variable explicative est connu, et tout ce qui concerne la structure de groupement et la distribution de la variable à expliquer est à décrire par le modèle de groupement. Un modèle de groupement standard est alors entièrement caractérisé par le choix des paramètres $I, \{i(v)\}_{1\leq v\leq V}, \{N_{ij}\}_{1\leq i\leq I, 1\leq j\leq J}$.

Afin de sélectionner le meilleur modèle connaissant les données de l'échantillon, on applique une approche Bayesienne similaire au cas de la discrétisation. La définition 2.4 présente une distribution a priori des modèles de groupement exploitant la hiérarchie "naturelle" des paramètres des modèles.

Définition 2.4. On appelle a priori hiérarchique l'a priori de modèle SGM basé sur les hypothèses suivantes :
- le nombre de groupes est compris entre 1 et V de façon équiprobable,
- pour un nombre de groupe donné I, toutes les partitions des V valeurs à expliquer en I groupes sont équiprobables,
- pour un groupe donné, toutes les distributions des valeurs de la variable à expliquer sont équiprobables,
- les distributions des valeurs de la variable à expliquer sur chaque groupe sont indépendantes les unes des autres.

En utilisant la définition formelle des modèles SGM et de leur distribution a priori hiérarchique, la formule de Bayes permet de calculer de manière exacte la probabilité d'un modèle connaissant les données, ce qui conduit au théorème 2.2, démontré dans [Boullé, 2005a] (article reproduit en annexe B : théorème 1 de la section 2.3).

Théorème 2.2. *Un modèle de groupement standard suivant un a priori hiérarchique est optimal au sens de Bayes si son évaluation par la formule suivante est minimale sur l'ensemble de tous les modèles :*

$$\log V + \log B\left(V, I\right) + \sum_{i=1}^{I} \log \binom{N_{i.} + J - 1}{J - 1} + \sum_{i=1}^{I} \log \frac{N_{i.}!}{N_{i1}! N_{i2}! \ldots N_{iJ}!}. \tag{2.7}$$

$B(V, I)$ est le nombre de répartitions des V valeurs explicatives en I groupes (éventuellement vides). Pour $I = V$, $B(V, I)$ correspond au nombre de Bell. Dans le cas général, $B(V, I)$ peut s'écrire comme une somme de nombre de Stirling de deuxième espèce (nombre de partitions de V valeur en i groupes non vides) :

$$B(V, I) = \sum_{i=1}^{I} S(V, i). \tag{2.8}$$

Le premier terme de la formule 2.7 correspond au choix du nombre de groupes et le second terme au choix de la partition des valeurs explicatives. Le troisième terme représente le choix des distributions de la variable à expliquer dans chaque groupe et le dernier terme la probabilité d'observer les valeurs de la variable à expliquer connaissant le modèle de groupement.

A titre d'exemple, le groupement de valeur optimal de la variable couleur de chapeau de la base de champignons est présenté sur la figure 2.3.

2.1.3 Grille bivariée

Les méthodes univariées de discrétisation ou groupement de valeurs sont limitées à une seule variable explicative. On propose dans cette section d'étendre ces méthodes au cas bivarié.

Les variables explicatives sont partitionnées, en intervalles dans le cas numérique et groupes de valeurs dans le cas catégoriel. La grille de données résultante permet alors d'évaluer la corrélation entre la paire de variables explicatives et la variable à expliquer. Le meilleur partitionnement bivarié est recherché au moyen d'une approche Bayesienne de la sélection de modèles.

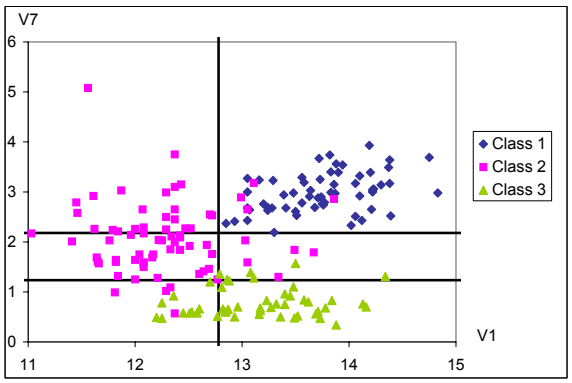
Après avoir introduit le problème au moyen d'un exemple illustratif, nous présentons un critère d'évaluation pour les modèles de grilles de données dans le cas de l'analyse bivariée pour la classification supervisée.

2.1.3.1 Présentation

La figure 2.4 présente le diagramme de dispersion des variables V1 et V7 du jeu de données Wine [Blake and Merz, 1996], catégorisé par valeur à expliquer. Chaque variable isolément est faiblement discriminante. La variable V1 ne peut séparer les classes 1 et 3 au delà de la valeur 13. De même, la variable V7 confond les classes 1 et 2 au-delà de la valeur 2. Les deux variables conjointement autorisent une meilleure discrimination des classes.

L'approche utilisée pour qualifier l'information prédictive contenue dans la paire de variables repose sur un partitionnement des individus en une grille de données. Chaque variable explicative est partitionnée en un ensemble de *parties* (intervalles dans le cas numérique et groupes de valeurs dans le cas catégoriel). Le produit cartésien des deux partitions univariées répartit les individus sur une *grille de données*, dont les *cellules* sont définies par des couples de parties. Le lien avec la variable à expliquer se fait au moyen de la distribution des valeurs à expliquer dans chaque cellule.

Par exemple dans la figure 2.4, la variable V1 est discrétisée en 2 intervalles (borne 12.78) et la variable V7 en 3 intervalles (bornes 1.235 et 2.18). Les individus se répartissent dans les 6 cellules de la grille bidimensionnelle ainsi définie. Dans chaque cellule, nous obtenons une distribution des valeurs à expliquer. Par exemple, le tableau sur la droite de la figure 2.4 montre que 63 individus ont abouti dans la cellule définie par les intervalles $]12.78, +\infty[$ sur V1 et $]2.18, +\infty[$ sur V7. Ces 63 individus sont distribués en 59 individus sur la classe 1 et 4 individus sur la classe 2.



V7xV1	]-∞, 12.78]	]12.78, +∞[
]2.18, +∞[	(0, 23, 0)	(59, 0, 4)
]1.235, 2.18]	(0, 35, 0)	(0, 5, 6)
]-∞, 1.235]	(0, 4, 11)	(0, 0, 31)

FIG. 2.4 – Diagramme de dispersion des variables explicatives V1 et V7 catégorisé par classe pour la base de données Wine. Une discrétisation 2D en six cellules est représentée sur le diagramme de dispersion et le triplet d'effectif par classe à expliquer est résumé par cellule de la grille de discrétisation 2D dans le tableau de droite.

Les grilles de données sont d'autant plus fiables qu'elles contiennent plus d'individus par cellule, et d'autant plus informatives que les cellules permettent de bien discriminer les valeurs à expliquer. Dans notre exemple, la grille de données optimale est tracée sur le diagramme de dispersion sur la gauche de la figure 2.4.

2.1.3.2 Formalisation dans le cas de deux variables numériques

Nous utilisons ici la même approche que dans le cas univarié pour rechercher le meilleur compromis entre information et fiabilité, en introduisant une famille de modèles de partitionnement bivarié, puis en choisissant le meilleur modèle au moyen d'une approche Bayesienne. Nous nous intéressons d'abord au cas des variables explicatives numériques, avant de généraliser à tous types de paires de variables, catégorielles ou mixtes.

Définition 2.5. Un modèle de partitionnement bivarié supervisé est défini par une partition en intervalles pour chaque variable explicative, et par la distribution des valeurs

à expliquer dans chaque cellule de la grille de données déduite du produit cartésien des deux partitions univariées.

Notations 2.3.
- X_1, X_2 : variables explicatives numériques
- N : nombre d'individus de l'échantillon
- J : nombre de valeurs de la variable à expliquer (connu)
- I_1, I_2 : nombre d'intervalles pour chaque variable explicative (inconnu)
- $N_{i_1..}$: nombres d'individus de l'intervalle i_1 de la variable X_1
- $N_{.i_2.}$: nombres d'individus de l'intervalle i_2 de la variable X_2
- $N_{i_1 i_2.}$: nombre d'individus de la cellule (i_1, i_2) des variables (X_1, X_2)
- $N_{i_1 i_2 j}$: nombre d'individus de la cellule (i_1, i_2) pour la valeur à expliquer j

Un modèle de partitionnement bivarié supervisé décrit la distribution des valeurs à expliquer, connaissant les valeurs explicatives. Il est entièrement défini par les nombres d'intervalles I_1 et I_2, les bornes des intervalles $\{N_{i_1..}\}$ et $\{N_{.i_2.}\}$ et la distribution des valeurs à expliquer $\{N_{i_1 i_2 j}\}$ par cellule (i_1, i_2) de la grille de données. Il est à noter que les nombres d'individus $\{N_{i_1 i_2.}\}$ par cellule de la grille ne font pas partie des paramètres du modèle : ils sont déduits du jeu de données à partir de la partition de chaque variable explicative.

Chaque information explicative est utilisée pour définir la famille de modèles. Les bornes des partitions univariées proviennent des valeurs des variables explicatives et les effectifs des cellules de la grille sont déduites de l'échantillon de données. Dans ce sens, les modèles en grille dépendent des données explicatives. Ce qui est décrit dans la famille de modèles est l'interaction entre les variables explicatives et la variable à expliquer.

Nous introduisons dans la définition 2.6 un a priori sur la distribution des paramètres des modèles de partitionnement bivarié supervisé, exploitant la hiérarchie des paramètres.

Définition 2.6. L'a priori hiérarchique sur l'espace des modèles de partitionnement bivarié supervisé est défini de la façon suivante :
- les nombres d'intervalles sont indépendants entre eux, et compris entre 1 et N de façon équiprobable,
- pour chaque variable explicative, pour un nombre d'intervalle donné, toutes les partitions en intervalles sur les rangs de la variable explicative sont équiprobables,
- pour chaque cellule de la grille de données, toutes les distributions des valeurs de la variable à expliquer sont équiprobables,
- les distributions des valeurs de la variable à expliquer sur chaque cellule sont indépendantes entre elles.

L'application de l'approche Bayesienne de la sélection de modèles conduit ici au critère d'évaluation d'un partitionnement bivarié supervisé, fourni dans la formule 2.9.

$$\begin{aligned} &\log N + \log \binom{N + I_1 - 1}{I_1 - 1} + \log N + \log \binom{N + I_2 - 1}{I_2 - 1} \\ &+ \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \log \binom{N_{i_1 i_2.} + J - 1}{J - 1} + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \log \frac{N_{i_1 i_2.}!}{N_{i_1 i_2 1}! N_{i_1 i_2 2}! \dots N_{i_1 i_2 J}!} \end{aligned} \tag{2.9}$$

De façon similaire au cas de la discrétisation univariée, les deux premiers termes d'a priori correspondent au choix de la partition (nombre d'intervalles et bornes) de la première variable explicative. De même, les deux termes suivants correspondent au choix de la partition de la deuxième variable explicative. Le dernier terme d'a priori, en début de la deuxième ligne, représente le choix de la distribution des valeurs à expliquer dans chaque cellule. Le dernier terme de la formule 2.9, sur la deuxième ligne, représente la vraisemblance, c'est à dire la probabilité d'observer les valeurs de la variable à expliquer dans les cellules de la grille connaissant le modèle de partitionnement bivarié supervisé.

2.1.3.3 Généralisation aux autres types de paires de variables

Dans le cas de deux variables catégorielles explicatives X_1 et X_2 comportant V_1 et V_2 valeurs, on applique la même approche. La variable X_1 est partitionnée en I_1 groupes de valeurs (au lieu d'intervalles dans le cas numérique) et la variable X_2 en I_2 groupes de valeurs. La distribution des valeurs à expliquer est décrite dans chaque cellule de la grille de données résultant du produit cartésien des partitions univariées. Comparativement au cas numérique, la seule modification a lieu dans la distribution a priori de chaque partition univariée. L'impact dans la formule 2.9 est de remplacer les termes relatifs à la distribution a priori d'une discrétisation en intervalles (deux premiers termes de la formule 2.4 de la discrétisation supervisée univariée) par les termes correspondant dans le cas d'un groupement de valeurs (deux premiers termes de la formule 2.4 du groupement de valeur supervisé univarié). Le critère d'évaluation résultant dans le cas de deux variables explicatives catégorielles est donné par la formule 2.10.

$$\begin{aligned}&\log V_1 + \log B\left(V_1, I_1\right) + \log V_2 + \log B\left(V_2, I_2\right)\\&+ \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2} \log \binom{N_{i_1 i_2 .} + J - 1}{J-1} + \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2} \log \frac{N_{i_1 i_2 .}!}{N_{i_1 i_2 1}! N_{i_1 i_2 2}! \ldots N_{i_1 i_2 J}!}\end{aligned} \tag{2.10}$$

Dans le cas mixte d'une variable catégorielle X_1 comportant V_1 valeurs et d'une variable numérique X_2, la première variable est partitionnée en groupes de valeurs et la seconde en intervalles. Le critère d'évaluation résultant dans le cas mixte est donné par la formule 2.11.

$$\begin{aligned}&\log V_1 + \log B\left(V_1, I_1\right) + \log N + \log \binom{N + I_2 - 1}{I_2 - 1}\\&+ \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2} \log \binom{N_{i_1 i_2 .} + J - 1}{J-1} + \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2} \log \frac{N_{i_1 i_2 .}!}{N_{i_1 i_2 1}! N_{i_1 i_2 2}! \ldots N_{i_1 i_2 J}!}\end{aligned} \tag{2.11}$$

2.1.4 Grille multivariée

On considère K variables explicatives $X_1, X_2, \ldots, X_K$, numériques ou catégorielles, et une variable à expliquer Y catégorielle. L'objectif d'un modèle d'estimation de densité conditionnelle est de décrire la valeur à expliquer des individus connaissant leur rang

pour les variables numériques explicatives et leur valeur pour les variables catégorielles explicatives.

Après avoir montré que la généralisation du bivarié au multivarié peut s'interpréter comme une sélection de variables, on présente le critère d'évaluation des grilles de données dans le cas multivarié supervisé.

2.1.4.1 Sélection de variables pour les grilles

Cette section introduit les enjeux de la sélection de variables pour les modèles en grille.

Sélection univariée et multivariée. Dans le cas la discrétisation univariée (ou du groupement de valeurs univarié), l'absence d'information discriminante se caractérise par une partition univariée de la variable explicative en un seul intervalle (cf. formule 2.4 du modèle nul $M_\emptyset$ de discrétisation). La discrétisation supervisée peut alors s'interpréter comme une sélection univariée, la variable étant éliminée si sa discrétisation comporte un seul intervalle et sélectionnée sinon.

Dans le cas multivarié, toute variable explicative partitionnée en une seule partie peut être considérée comme éliminée, les variables restantes étant sélectionnées. Dans ce cadre, la sélection de variables sera d'autant plus performante que les variables explicatives indépendantes de la variable à expliquer seront éliminées.

Risque lié aux grandes dimensions. Le risque de sélection à tort d'une variable indépendante croît avec le nombre de variables explicatives. En effet, en ajoutant indéfiniment des variables indépendantes dans la représentation des données, la probabilité qu'au moins une des variables explicatives présente des corrélations avec la variable à expliquer tend vers 1.

Prenons par exemple un échantillon de données de taille $N = 20$ comportant une seule variable numérique explicative et une variable à expliquer booléenne à valeurs 0 et 1. Si l'échantillon ordonné selon la variable explicative se décompose en une séqunece de 0 suivi d'une séquence de 1, la discrétisation de cette variable en deux intervalles purs (ne contenant que des 0 ou que des 1) est probablement pertinente. Si par contre un million de variables explicatives sont évaluées, la probabilité qu'au moins une d'entre elles permette "par hasard" un découpage des valeurs explicatives en deux intervalles purs devient significative. En effet, puisque $2^{20} \approx 10^6$, pratiquement toutes les combinaisons possibles des 20 valeurs à expliquer peuvent être observées.

Comparaison de grilles de différentes dimensions. Le problème de la sélection de variables se pose déjà dans les cas univarié et bivarié. Il n'a pas été introduit précédemment essentiellement pour des raisons didactiques de progression dans la présentation des modèles en grille. Il est nécessaire de le prendre en compte dès lors que l'on veut comparer des grilles comportant des nombres différents de variables.

Par exemple, pour K variables explicatives, les modèles en grille sont au nombre de K dans le cas univarié, $K(K-1)/2$ dans le cas bivarié et $K(K-1)(K-2)/6$ dans le cas de trois variables explicatives. Il s'agit de pouvoir comparer l'apport informatif d'une

grille univariée et d'une grille multivariée, en tenant compte d'une part de la qualité de la discrimination des valeurs à expliquer, d'autre part du nombre de modèles en grille évalués.

Enrichissement du paramétrage des grilles dans le cas multivarié. Afin de gérer le risque des grandes dimensions et de permettre la comparaison de modèles en grille quelle que soit leur dimension, on va explicitement considérer le passage au multivarié comme un problème de sélection de variables, en incorporant le choix du nombre de variables et du sous-ensemble des variables sélectionnées comme un niveau supplémentaire dans la hiérarchie du paramétrage des modèles en grille de données.

2.1.4.2 Distribution a priori pour la sélection de variables

Comme précédemment, on propose d'utiliser une distribution a priori exploitant la hiérarchie des paramètres, en choisissant d'abord le nombre de variables sélectionnées, puis le sous-ensemble des variables sélectionnées.

Pour un nombre K de variables explicatives, on choisit le nombre K_s de variables sélectionnées selon un a priori uniforme entre 0 et K variables, représentant $K+1$ alternatives équiprobables.

Pour le choix des K_s variables, on affecte la même probabilité a priori à chaque sous-ensemble de K_s variables. Le nombre de combinaisons $\binom{K}{K_s}$ semble naturel pour cet a priori, mais il a le désavantage d'être symétrique. Au delà de $K/2$ variables, chaque nouvelle variable rend la sélection plus probable. Ainsi, l'ajout de variables inutiles est favorisé, pourvu que l'impact sur la vraisemblance du modèle soit négligeable. Comme on préfère les modèles simples aux modèle complexes, on propose d'utiliser le nombre de combinaisons avec remise $\binom{K+K_s-1}{K_s}$.

En prenant le log négatif de cet a priori, on obtient la longueur de codage [Shannon, 1948] suivante pour la sélection de variables :

$$\log(K+1) + \log \binom{K+K_s-1}{K_s}. \tag{2.12}$$

En utilisant cet a priori, le "coût informationnel" de sélection est croissant avec le nombre de variables. Les premières variables sélectionnées coûtent environ $\log K$ et les dernières environ $\log 2$.

2.1.4.3 Formalisation

On recherche un partitionnement de chaque variable explicative, en intervalles pour les variables numériques et en groupes de valeurs pour les variables catégorielles. La taille des partitions détermine les dimensions d'une grille explicative dans laquelle les individus sont affectés. La définition 2.7 formalise la description d'un modèle de classification par grille, en incorporant explicitement la sélection de variables dans la structure du paramétrage.

Définition 2.7. Un modèle de classification par grille est défini par :

– un nombre de variables explicatives sélectionnées,

- un sous-ensemble de variables explicatives sélectionnées,
- un nombre d'intervalles (ou groupes) pour chaque variable explicative sélectionnée,
- une partition des valeurs en intervalles (ou groupes) pour chaque variable explicative sélectionnée,
- pour chaque cellule de la grille explicative de données ainsi définie, la distribution des valeurs à expliquer.

Notations 2.4.

- N : nombre d'individus de l'échantillon
- J : nombre de valeurs de la variable à expliquer (connu)
- K : nombre de variables explicatives
- $\mathbb{K}$: ensemble des variables explicatives ($|\mathbb{K}| = K$)
- $\mathbb{K}_1$: sous-ensemble des variables explicatives de type numérique
- $\mathbb{K}_2$: sous-ensemble des variables explicatives de type catégoriel
- K_s : nombre de variables explicatives sélectionnées (inconnu)
- $\mathbb{K}_s$: sous-ensemble des variables explicatives sélectionnées ($|\mathbb{K}_s| = K_s$)
- $V_k, k \in \mathbb{K}_2$: nombre de valeurs de la variable catégorielle X_k
- I_k : taille de la partition univariée (en intervalles ou groupes) de la variable X_k (inconnu)
- $N_{i_1 i_2 \ldots i_K}$: nombre d'individus de la cellule $(i_1, i_2, \ldots, i_K)$
- $N_{i_1 i_2 \ldots i_K j}$: nombre d'individus de la cellule $(i_1, i_2, \ldots, i_K)$ pour la valeur à expliquer j

On exploite la structure des paramètres des modèles pour hiérarchiser les décisions de choix des paramètres en étant uniforme à chaque niveau de la hiérarchie, ce qui induit une distribution a priori des modèles.

On commence par choisir le nombre de variables sélectionnées, puis le sous-ensemble des variables sélectionnées. On détermine ensuite les dimensions de la grille en choisissant le nombre de groupes ou d'intervalles, indépendamment pour chaque variable explicative sélectionnée. Les partitions univariées sont alors spécifiées, en groupes de valeurs dans le cas catégoriel et en intervalles dans le cas numérique. Ces partitionnements par variable permettent de peupler la grille de données, en exploitant les rangs et valeurs explicatives qui sont connus. Ceci détermine l'effectif par cellule de la grille explicative. Il faut maintenant décrire pour chaque cellule la distribution des individus sur les valeurs à expliquer, ce qui est effectué au moyen d'un modèle multinômial. Enfin, la vraisemblance des valeurs à expliquer observées dans l'échantillon est évaluée localement à chaque cellule selon la formule du multinôme.

La démarche décrite ci-dessus permet de calculer exactement la probabilité d'un modèle de classification par grille connaissant les données, ce qui conduit au théorème 2.3.

Théorème 2.3. *Un modèle de classification par grille suivant un a priori hiérarchique est optimal au sens de Bayes si son évaluation par la formule suivante est minimale sur*

l'ensemble de tous les modèles :

$$\begin{aligned}
&\log(K+1) + \log \binom{K+K_s-1}{K_s} \\
&+ \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_1} \log N + \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_2} \log V_k \\
&+ \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_1} \log \binom{N+I_k-1}{I_k-1} + \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_2} \log B(V_k, I_k) \\
&+ \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_K=1}^{I_K} \log \binom{N_{i_1 i_2 \dots i_K} + J - 1}{J-1} \\
&+ \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_K=1}^{I_K} \left(\log N_{i_1 i_2 \dots i_K}! - \sum_{j=1}^{J} \log N_{i_1 i_2 \dots i_K j}! \right)
\end{aligned} \tag{2.13}$$

La première ligne du critère d'évaluation correspond au choix du nombre de variables sélectionnées et au choix du sous-ensemble de variables. La deuxième ligne correspond au choix du nombre d'intervalles (ou groupes) pour les variables explicatives sélectionnées et la troisième ligne au choix de leur partition en intervalles (ou groupes de valeurs). La quatrième ligne du critère représente le choix de la distribution des valeurs à expliquer pour chaque cellule de la grille explicative. La dernière ligne correspond à la probabilité d'observer les valeurs de la variable à expliquer connaissant le modèle en grille (formule du multinôme pour chaque cellule).

Pour des raisons de simplification de la notation, les deux dernières lignes (K signes sommes imbriqués) prennent en compte toutes les variables sélectionnées ou non. En fait, les signes sommes correspondant aux variable non sélectionnées (partition en une seule partie) sont inutiles.

On peut noter que dans le cas où aucune variable n'est sélectionnée, le modèle en grille ne contient plus qu'une seule cellule. La formule 2.13 se réduit alors à

$$\log(K+1) + \log \binom{N+J-1}{J-1} + \log \frac{N!}{N_1! N_2! \dots N_J!} \tag{2.14}$$

en notant N_j l'effectif de la valeur à expliquer j. Cela correspond comme dans 2.5 à la probabilité a posteriori d'un modèle multinômial des valeurs à expliquer, indépendant des variables explicatives.

2.2 Régression

Dans un problème de régression supervisée tel que formulé classiquement, l'échantillon de données est constitué de N individus, K variables explicatives $X_1, X_2, \dots, X_K$ numériques et d'une variable à expliquer numérique Y. On cherche à modéliser la variable à expliquer comme une fonction des variables explicatives, sous la forme $Y = f(X_1, X_2, \dots, X_K)$.

On s'intéresse ici à l'expression de la dépendance entre la variable à expliquer et les variable explicatives sous une forme probabiliste $P(Y|X_1, X_2, \ldots, X_K)$.

Dans cette section, on introduit un modèle d'estimation de densité conditionnelle dans le cas univarié d'une variable explicative numérique au moyen d'une discrétisation bivariée conjointement de la variable explicative et de la variable à expliquer. On étend ce modèle au cas univarié d'une variable explicative catégorielle, avant de présenter sa généralisation au cas multivarié pour un nombre quelconque de variables explicatives.

2.2.1 Discrétisation

La variable explicative considérée est ici numérique. Après avoir introduit le problème au moyen d'un exemple illustratif, on propose un modèle d'estimation de densité conditionnelle basé sur la discrétisation à la fois de la variable explicative et de la variable à expliquer, puis un critère d'évaluation de la qualité d'un modèle fondé sur une approche Bayesienne.

2.2.1.1 Présentation

La base Iris de Fisher est ici utilisée en prenant la longueur de pétale comme variable explicative et la longueur de sépale comme variable à expliquer, ce qui correspond à un problème de régression. La figure 2.5 reporte sur un diagramme de dispersion chaque individu de l'échantillon de données en prenant en abscisse la variable explicative et en ordonnée la variable à expliquer.

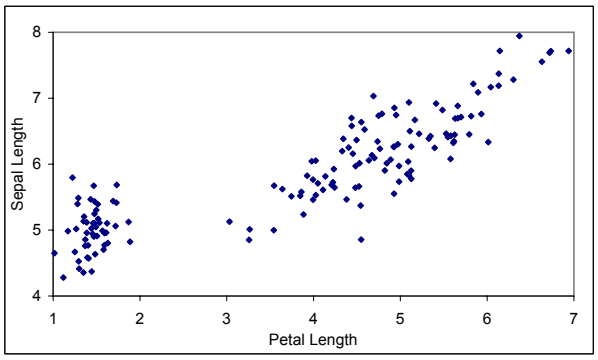


FIG. 2.5 – Diagramme de dispersion des longueurs de sépale en fonction des longueurs de pétale dans la base Iris.

En visualisant ce diagramme de dispersion, la variable explicative longueur de pétale paraît clairement informative pour la prédiction de la longueur de sépale. Cette analyse visuelle, bien qu'intéressante, n'est pas applicable si l'on devait évaluer l'information prédictive pour de très nombreuses variables explicatives : il n'est pas possible de comparer visuellement des centaines de diagrammes de dispersion. De plus, des problèmes d'échelle peuvent entraver la lisibilité des diagrammes de dispersion. Il serait intéressant de quantifier l'information prédictive de la variable explicative.

On propose d'estimer l'information prédictive contenue dans la variable explicative au moyen d'un modèle de discrétisation 2D des variables. Autrement dit, on partitionne à la fois les longueurs de pétale et de sépale en intervalles. Ces deux discrétisations permettent de découper le diagramme de dispersion en une grille de cellules. La proportion d'iris appartenant à chaque cellule de la grille est estimée simplement par comptage.

La figure 2.6 montre deux exemples de grilles, comportant $6 = 3 * 2$ cellules ou $96 = 12 * 8$ cellules. La grille en 6 cellules correspond à une discrétisation des longueurs de pétales en trois intervalles et des longueurs de sépale en 2 intervalles. Quand la longueur de pétale est plus petite que 3 cm, 100% des iris ont une longueur de sépale inférieure à 6 cm. A l'opposé, pour les longueurs de pétale supérieure à 5 cm, environ 90% des iris ont une longueur de sépale supérieure à 6 cm. Pour les longueurs de pétale intermédiaire (entre 3 et 5 cm), toutes les longueurs de sépale sont également possibles (proportion 53%-47% sur les deux intervalles à expliquer).

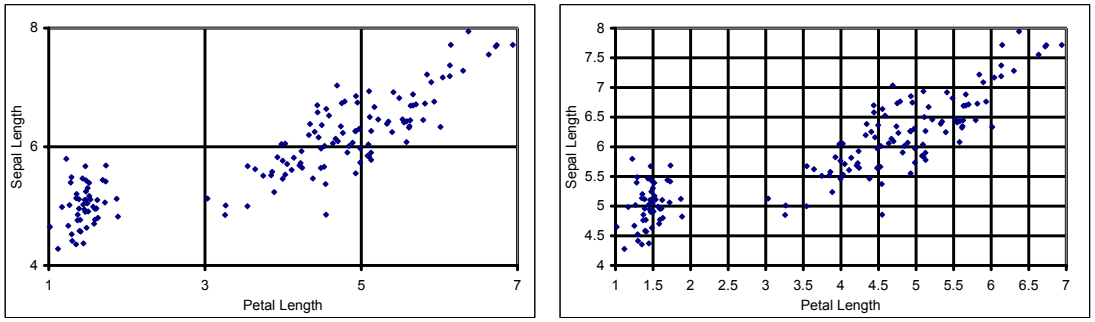


FIG. 2.6 – Découpage en 6 ou 96 cellules du diagramme de dispersion des longueurs de sépale et de pétale de la base Iris.

Cette analyse est informative, mais la grille utilisée est elle la meilleure possible ? La grille en 96 cellules est beaucoup plus précise et permet de faire des prédictions très fines sur la longueur de sépale connaissant la longueur de pétale. On peut néanmoins se poser la question de la robustesse de telles prédictions. Jusqu'à quel point peut-on faire confiance à une prédiction du type "100% des iris ayant une longueur de pétale comprise entre 6.5 et 7.0 cm ont une longueur de sépale comprise entre 7.5 et 8 cm" ?

L'utilisation de modèles de grille permet de passer d'une grille mono-cellule, exprimant l'absence d'information prédictive, à une grille ayant au plus un point par cellule, ce qui correspond manifestement à une sur-discrétisation des variables explicative et à expliquer. Le problème de choix du bon modèle est alors critique :

- Quel est le bon nombre d'intervalles pour chaque variable ?
- Comment choisir les bornes des intervalles ?
- Comment caractériser l'absence d'information discriminante ?

2.2.1.2 Formalisation

Soit X une variable explicative numérique, Y une variable à expliquer numérique et $D = \{(x_n, y_n); 1 \leq n \leq N\}$ un échantillon de données (X, Y).

Habituellement dans les problèmes de régression, on recherche une fonction permettant d'exprimer la variable à expliquer sous la forme $Y = f(X)$ ou $Y = f(X) + \varepsilon$ si l'on tient compte du bruit dans les données. On recherche ici un modèle d'estimation de la densité conditionnelle $P(Y|X)$ de Y sachant X.

Comme dans le choix 2.1 introduit pour la discrétisation supervisée, notre parti pris est qu'une bonne estimation de densité conditionnelle $P(Y|X)$ doit être invariante par toute transformation monotone des variables explicative ou à expliquer et peu sensible aux valeurs atypiques (outliers). On se base sur la statistique d'ordre en s'intéressant aux rangs plutôt qu'aux valeurs. Il s'agit donc en fait d'estimer le rang de Y conditionnellement au rang de X.

Il n'est pas réaliste de viser une prédiction exacte de chaque rang de la variable à expliquer, ceux ci étant en trop grand nombre. On propose de modéliser les rangs à expliquer par intervalles de rangs, ce qui permet de se ramener à un problème de classification supervisée, à savoir l'estimation des intervalles de rangs de la variable à expliquer conditionnellement aux rangs de la variable explicative, puis à la spécification des rangs à expliquer localement à chaque intervalle à expliquer.

La définition 2.8 formalise cette approche pour décrire une famille de modèles d'estimation de densité conditionnelle, pour laquelle la densité conditionnelle $P(Y|X)$ des intervalles à expliquer connaissant les intervalles explicatifs est constante par intervalle explicatif. Par facilité de langage, on parlera de modèle de régression par grille plutôt que de modèle d'estimation de densité conditionnelle.

Définition 2.8. Un modèle de régression par grille pour une variable explicative numérique est défini par :

- un nombre d'intervalles pour les variables explicative et à expliquer,
- une partition de la variable explicative en intervalles, spécifiée sur les rangs des valeurs explicatives,
- pour chaque intervalle explicatif, la distribution des individus sur les intervalles à expliquer, spécifiée par les effectifs de chaque intervalle à expliquer localement à l'intervalle explicatif.

Notations 2.5.

- N : nombre d'individus de l'échantillon
- I : nombre d'intervalles explicatifs (inconnu)
- J : nombre d'intervalles à expliquer (inconnu)
- $N_{i.}$: nombre d'individus de l'intervalle explicatif i
- $N_{.j}$: nombre d'individus de l'intervalle à expliquer j
- N_{ij} : nombre d'individus de l'intervalle explicatif i pour l'intervalle à expliquer j

Un modèle de régression par grille est entièrement caractérisé par le choix des paramètres $I, J, \{N_{i.}\}_{1 \leq i \leq I}, \{N_{ij}\}_{1 \leq i \leq I, 1 \leq j \leq J}$.

De façon similaire au cas de la classification supervisée, la distribution des individus $\{N_{ij}\}$ sur les intervalles à expliquer est spécifiée localement à chaque intervalle explicatif. De ce fait, les effectifs $N_{.j}$ des intervalles à expliquer ne sont pas explicitement modélisés : ils sont déduits pour chaque intervalle à expliquer par sommation des effectifs N_{ij} par cellule.

Afin d'obtenir la relation globale entre les rangs de X et ceux de Y, il suffit de spécifier cette relation dans chaque intervalle à expliquer. Connaissant le rang d'un individu dans son intervalle à expliquer et le rang de cet intervalle dans la partition des intervalles à expliquer, le rang à expliquer global de l'individu s'en déduit immédiatement.

Afin de sélectionner le meilleur modèle connaissant les données de l'échantillon, on applique une approche Bayesienne similaire au cas de la classification supervisée. La définition 2.9 présente une distribution a priori des modèles de régression exploitant la hiérarchie "naturelle" des paramètres des modèles.

Définition 2.9. On appelle a priori hiérarchique l'a priori de modèle de régression par grille basé sur les hypothèses suivantes :

- les nombres d'intervalles explicatif et à expliquer sont indépendants, et compris entre 1 et N de façon équiprobable,
- pour un nombre d'intervalles explicatifs donné, toutes les partitions en intervalles des rangs de la variable explicative sont équiprobables,
- pour un intervalle explicatif donné, toutes les distributions des individus sur les intervalles à expliquer sont équiprobables,
- les distributions des intervalles à expliquer sur chaque intervalle explicatif sont indépendantes les unes des autres,
- pour un intervalle à expliquer donné, toutes les distributions des rangs des individus sont équiprobables.

En utilisant la définition formelle des modèles de régression par grille et de leur distribution a priori hiérarchique, la formule de Bayes permet de calculer de manière exacte la probabilité d'un modèle connaissant les données, ce qui conduit au théorème 2.4.

Théorème 2.4. *Un modèle de régression par grille suivant un a priori hiérarchique est optimal au sens de Bayes si son évaluation par la formule suivante est minimale sur l'ensemble de tous les modèles :*

$$\begin{aligned} &2\log N + \log \binom{N+I-1}{I-1} + \sum_{i=1}^{I} \log \binom{N_{i.}+J-1}{J-1} \\ &+ \sum_{i=1}^{I} \log \frac{N_{i.}!}{N_{i1}!N_{i2}!\dots N_{iJ}!} + \sum_{j=1}^{J} \log N_{.j}! \end{aligned} \tag{2.15}$$

Par rapport au critère obtenu dans le cas de la classification supervisée, il y a un terme supplémentaire égal à $\log(N)$ pour la prise en compte du choix du nombre d'intervalles à expliquer selon la loi a priori et un terme additif en $\log N_{.j}!$ qui évalue la vraisemblance de la distribution des rangs des individus dans chaque intervalle à expliquer.

Notons que dans le cas d'une variable explicative indépendante de la variable à expliquer, le modèle de discrétisation optimal ne comporte qu'un intervalle explicatif et à expliquer. La formule 2.15 se reduit alors à

$$2\log N + \log N!, \tag{2.16}$$

ce qui correspond essentiellement à $N!$ possibilités pour spécifier le rangement des valeurs à expliquer des N individus.

2.2.2 Groupement de valeurs

Toujours dans le cadre de la régression, la variable explicative considérée est ici catégorielle. L'approche classique utilisée en régression consiste à recoder la variable catégorielle en numérique, par exemple par codage disjonctif complet (création d'une variable numérique booléenne par valeur explicative). Cette approche rencontre ses limites quand le nombre de valeurs explicatives devient important. On propose ici un modèle d'estimation de la densité conditionnelle par grille, basée sur le groupement de la variable explicative et la discrétisation de la variable à expliquer.

2.2.2.1 Présentation

La base Adult de l'UCI [Blake and Merz, 1996] est issue d'un recensement aux Etats-Unis. Elle comporte environ 50000 individus pour une quinzaine de variables. On s'intéresse ici à la variable workclass en tant que variable explicative et à la variable âge comme variable à expliquer. La figure 2.7 présente l'histogramme par tranche d'âge pour chaque valeur de workclass.

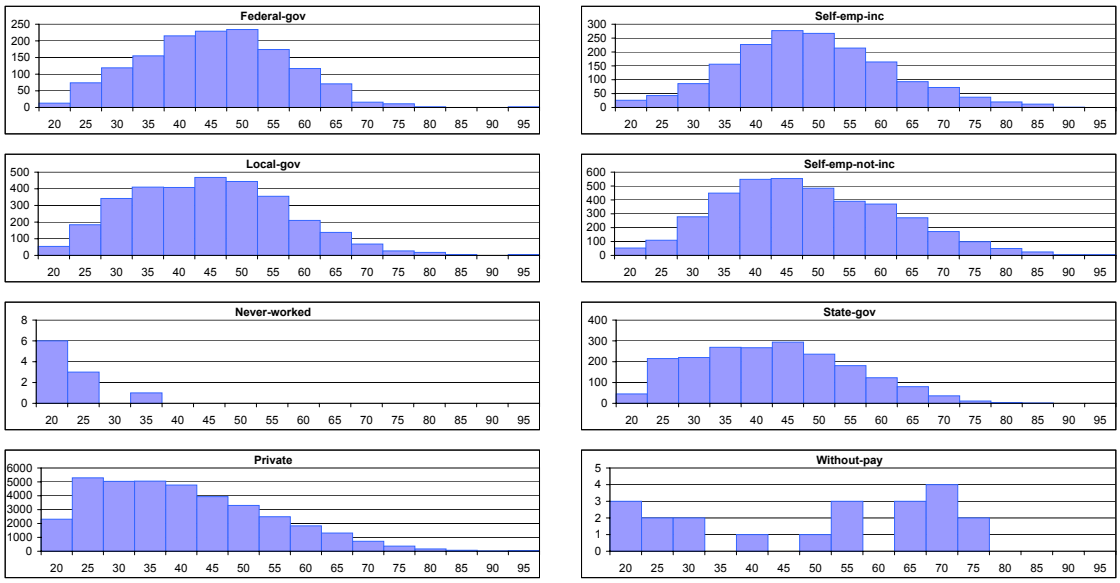


FIG. 2.7 – Histogrammes par tranche d'âge pour chacune des valeurs de la variable explicative workclass, dans la base Adult.

Connaissant la valeur de la variable explicative workclass, les distributions par âge paraissent discernables. Par exemple, pour la valeur Private, l'effectif maximum est atteint pour la tranche d'âge autour de 25 ans, alors qu'il se situe autour de 50 ans pour la valeur Federal-gov. Les valeurs Federal-gov, Self-emp-inc, Local-gov et Self-emp-not-inc exhibent des histogrammes similaires : on est tenté de les regrouper ensemble. La valeur State-gov a un comportement intermédiaire entre celui de ce groupe et celui de la valeur Private : il est difficile de savoir si ce comportement doit être isolé ou au contraire rapproché d'un des deux comportements précédemment identifiés. Les deux dernières valeurs Never-worked

et Without-pay sont atypiques, mais leur effectif est si faible que l'on peut se demander si les différences observées sont significatives.

En ce qui concerne la variable à expliquer âge, il faut déterminer le niveau de finesse des prédictions que l'on peut raisonnablement effectuer. Doit-on se limiter à deux ou trois tranches d'âge, en privilégiant la robustesse des prédictions, ou au contraire utiliser plus d'une vingtaine de tranches pour prédire la distribution des âges de façon très précise ? A titre d'exemple, le tableau 2.2 présente les distributions des tranches d'âge conditionnellement aux valeurs de workclass, dans le cas d'une partition de la variable explicative workclass en trois groupes et d'un découpage de la variable à expliquer âge en 6 intervalles.

TAB. 2.2 – Distribution des individus de la base Adult par tranche d'âge conditionnellement aux valeurs de workclass, pour une grille bidimensionnelle en trois groupes de valeurs workclass et six intervalles de la variable âge à expliquer. Par exemple, en se basant sur la première case du tableau, 1.4% des 10125 individus du groupe de valeurs {Federal-gov, Local-gov, Self-emp-inc, Self-emp-not-inc} sont dans la tranche d'âge 0 à 20 ans.

workclass * âge	0-20	20-30	30-40	40-50	50-60	60-95	Effectif
Federal-gov, Local-gov, Self-emp-inc, Self-emp-not-inc	1.4%	12.2%	25.4%	29.2%	19.7%	12.1%	10125
State-gov	2.3%	22.0%	27.1%	26.8%	15.3%	6.6%	1981
Private, Never-worked, Without-pay	6.3%	28.1%	26.7%	19.7%	11.8%	7.3%	36736

On a ici introduit une famille de modèles d'estimation de la densité conditionnelle des âges connaissant la valeur de workclass. La variable explicative workclass est partitionnée groupes de valeurs et la variable à expliquer âge en intervalles, ce qui permet de décrire la distribution des tranches d'âge par groupe de valeurs sur la grille ainsi définie. Le problème du choix du modèle le plus pertinent se ramène ici principalement à déterminer la finesse de résolution de la grille.

2.2.2.2 Formalisation

Soit X une variable explicative catégorielle, Y une variable à expliquer numérique et $D = \{(x_n, y_n); 1 \leq n \leq N\}$ un échantillon de données (X, Y).

On recherche un modèle d'estimation de la densité conditionnelle $P(Y|X)$ de Y sachant X. De façon similaire au cas précédent, on propose un modèle en grille basé sur une partition en groupes de valeurs pour la variable explicative et en intervalles pour la variable à expliquer. La définition 2.10 formalise cette approche pour décrire la famille de modèles d'estimation de densité conditionnelle, dans le cas d'une variable explicative catégorielle.

Définition 2.10. Un modèle de régression par grille pour une variable explicative catégorielle est défini par :

– un nombre de groupes de valeurs pour la variable explicative et d'intervalles pour la variable à expliquer,

- une partition de la variable explicative en groupes de valeurs,
- pour chaque groupe de valeurs explicatif, la distribution des individus sur les intervalles à expliquer, spécifiée par les effectifs de chaque intervalle à expliquer localement au groupe de valeurs explicatif.

Notations 2.6.
- N : nombre d'individus de l'échantillon
- V : nombre de valeurs explicatives
- I : nombre de groupes explicatifs (inconnu)
- J : nombre d'intervalles à expliquer (inconnu)
- $i(v)$: index du groupe auquel est rattachée la valeur explicative v
- $N_{i.}$: nombre d'individus de l'intervalle explicatif i
- $N_{.j}$: nombre d'individus de l'intervalle à expliquer j
- N_{ij} : nombre d'individus de l'intervalle explicatif i pour l'intervalle à expliquer j

Un modèle de régression par grille est entièrement caractérisé par le choix des paramètres $I, J, \{i(v)\}_{1\leq v\leq V}, \{N_{ij}\}_{1\leq i\leq I, 1\leq j\leq J}$.

Afin de sélectionner le meilleur modèle connaissant les données de l'échantillon, on applique une approche Bayesienne similaire au cas de la classification supervisée. La définition 2.11 présente une distribution a priori des modèles de régression exploitant la hiérarchie "naturelle" des paramètres des modèles.

Définition 2.11. On appelle a priori hiérarchique l'a priori de modèle de régression par grille basé sur les hypothèses suivantes :
- le nombre de groupes explicatifs est compris entre 1 et V de façon équiprobable,
- le nombre d'intervalles à expliquer est indépendant du nombre de groupes, et compris entre 1 et N de façon équiprobable,
- pour un nombre de groupes explicatifs donné I, toutes les partitions des V valeurs explicatives en I groupes sont équiprobables,
- pour un groupe explicatif donné, toutes les distributions des individus sur les intervalles à expliquer sont équiprobables,
- les distributions des intervalles à expliquer sur chaque groupe explicatif sont indépendantes les unes des autres,
- pour un intervalle à expliquer donné, toutes les distributions des rangs des individus sont équiprobables.

En utilisant la définition formelle des modèles de régression par grille et de leur distribution a priori hiérarchique, la formule de Bayes permet de calculer de manière exacte la probabilité d'un modèle connaissant les données, ce qui conduit au théorème 2.5.

Théorème 2.5. *Un modèle de régression par grille suivant un a priori hiérarchique est optimal au sens de Bayes si son évaluation par la formule suivante est minimale sur l'ensemble de tous les modèles :*

$$\begin{aligned} &\log V + \log N + \log B(V, I) + \sum_{i=1}^{I} \log \binom{N_{i.} + J - 1}{J - 1} \\ &+ \sum_{i=1}^{I} \log \frac{N_{i.}!}{N_{i1}! N_{i2}! \dots N_{iJ}!} + \sum_{j=1}^{J} \log N_{.j}! \end{aligned} \tag{2.17}$$

La preuve du théorème 2.5 se déduit du groupement de valeurs d'une variable explicative catégorielle pour la classification supervisée (théorème 2.2) et de la régression par grille pour une variable à expliquer numérique (théorème 2.4).

2.2.3 Grille multivariée

Dans le cas général, on partitionne chaque variable explicative, en intervalles ou groupes de valeurs selon le type de la variable. La taille des partitions détermine les dimensions d'une grille explicative dans laquelle les individus sont affectés. L'approche est la même que dans le cas de la classification supervisée (section 2.1), exceptée que la variable numérique à expliquer est ici discrétisée en intervalles. Dans chaque cellule de la grille explicative, les individus sont distribués sur les intervalles à expliquer, au lieu d'être distribués sur les valeurs catégorielles à expliquer dans le cas de la classification supervisée. L'aspect sélection de variables est également incorporé dans le paramétrage des modèles de régression en grille.

La définition 2.12 formalise la description d'un modèle d'estimation de densité conditionnelle par grille dans le cas de la régression.

Définition 2.12. Un modèle de régression par grille est défini par :
- un nombre de variables explicatives sélectionnées,
- un sous-ensemble de variables explicatives sélectionnées,
- un nombre d'intervalles de la variable à expliquer,
- un nombre d'intervalles (ou groupes) pour chaque variable explicative sélectionnée,
- une partition des valeurs en intervalles (ou groupes) pour chaque variable explicative sélectionnée,
- pour chaque cellule de la grille explicative de données ainsi définie, la distribution des intervalles à expliquer.

Notations 2.7.
- N : nombre d'individus de l'échantillon
- K : nombre de variables explicatives
- $\mathbb{K}$: ensemble des variables explicatives ($|\mathbb{K}| = K$)
- $\mathbb{K}_1$: sous-ensemble des variables explicatives de type numérique
- $\mathbb{K}_2$: sous-ensemble des variables explicatives de type catégoriel
- K_s : nombre de variables explicatives sélectionnées (inconnu)
- $\mathbb{K}_s$: sous-ensemble des variables explicatives sélectionnées ($|\mathbb{K}_s| = K_s$)
- $V_k, k \in \mathbb{K}_2$: nombre de valeurs de la variable catégorielle X_k
- J : nombre d'intervalles de la variable à expliquer (inconnu)
- I_k : taille de la partition univariée (en intervalles ou groupes) de la variable X_k (inconnu)
- $N_{i_1 i_2 \ldots i_K}$: nombre d'individus de la cellule $(i_1, i_2, \ldots, i_K)$
- $N_{i_1 i_2 \ldots i_K j}$: nombre d'individus de la cellule $(i_1, i_2, \ldots, i_K)$ pour l'intervalle à expliquer j
- N_j : nombre d'individus de l'intervalle à expliquer j

Il est à noter que les nombres d'individus N_j par intervalle à expliquer ne font pas partie des paramètres d'un modèle de régression par grille : ils sont déduits sur chaque intervalle à expliquer par sommation sur les cellules.

Comme dans le cas de la classification supervisée, on exploite la structure des paramètres des modèles pour hiérarchiser les décisions de choix des paramètres en étant uniforme à chaque niveau de la hiérarchie, ce qui induit une distribution a priori des modèles.

La démarche décrite ci-dessus permet de calculer exactement la probabilité d'un modèle de régression par grille connaissant les données, ce qui conduit au théorème 2.6.

Théorème 2.6. *Un modèle de régression par grille suivant un a priori hiérarchique est optimal au sens de Bayes si son évaluation par la formule suivante est minimale sur l'ensemble de tous les modèles :*

$$\begin{aligned}
&\log(K+1) + \log \binom{K+K_s-1}{K_s} + \log N \\
&+ \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_1} \log N + \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_2} \log V_k \\
&+ \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_1} \log \binom{N+I_k-1}{I_k-1} + \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_2} \log B(V_k, I_k) \\
&+ \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_K=1}^{I_K} \log \binom{N_{i_1 i_2 \ldots i_K}+J-1}{J-1} \\
&+ \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_K=1}^{I_K} \left(\log N_{i_1 i_2 \ldots i_K}! - \sum_{j=1}^{J} \log N_{i_1 i_2 \ldots i_K j}! \right) \\
&+ \sum_{j=1}^{J_1} \log N_j!
\end{aligned} \tag{2.18}$$

Par rapport au critère obtenu dans le cas de la classification supervisée, il y a un terme supplémentaire égal à $\log(N)$ (en fin de première ligne du critère) pour la prise en compte du choix du nombre d'intervalles à expliquer selon la loi a priori et un terme additif en $\log N_j!$ (sur la dernière ligne du critère) qui évalue la vraisemblance de la distribution des rangs des individus dans chaque intervalle à expliquer.

On peut noter que dans le cas où aucune variable n'est sélectionnée, la formule 2.18 se réduit à

$$\log(K+1) + \log N + \log N!, \tag{2.19}$$

ce qui comme dans 2.16 correspond essentiellement à $N!$ possibilités pour spécifier le rangement des valeurs à expliquer des N individus.

2.3 Classification supervisée généralisée

On s'intéresse ici à nouveau au problème de la classification supervisée, en le généralisant au cas d'un nombre de valeurs à expliquer quelconque, non nécessairement petit comme cela est implicitement supposé habituellement. Après avoir introduit le problème, on étend les modèles de classification par grille présentés en section 2.1 en partitionnant la variable catégorielle à expliquer en groupes de valeurs.

2.3.1 Variable catégorielle à expliquer avec nombreuses valeurs

L'objectif de la classification est de prédire la valeur d'une variable catégorielle à expliquer connaissant l'ensemble des valeurs des variables explicatives. La plupart des problèmes de classification considérés usuellement se limitent à la prédiction d'une valeur booléenne, ou d'une variable comportant un nombre très faible de valeurs, typiquement moins de cinq valeurs.

On rencontre néanmoins des problèmes où ce nombre de valeurs à expliquer est plus important, comme par exemple la reconnaissance de chiffres manuscrits [LeCun *et al.*, 1995], la reconnaissance de caractères [Frey and Slate, 1991] ou la classification de texte issus de newsgroups [Joachims, 1997], qui comportent respectivement 10, 26 et 20 valeurs catégorielles à expliquer. De façon générale, pourquoi ne pas envisager le problème de classification dans son cadre le plus général sans faire l'hypothèse d'un nombre restreint de valeurs à expliquer ?

En présence d'un trop grand nombre de valeurs à expliquer, il n'est pas raisonnable de modéliser directement les valeurs. On propose de partitionner les valeurs catégorielles à expliquer en groupes de valeurs, ce qui permet de se ramener à un problème de classification supervisée standard portant sur un faible nombre de groupes de valeurs, puis de décrire la valeur effective de chaque individu localement à son groupe de valeurs.

2.3.2 Groupement des valeurs d'une variable catégorielle à expliquer

Soit Y une variable catégorielle à expliquer comportant V valeurs. Il s'agit d'étendre les modèles de classification supervisée en incorporant un groupement des V valeurs en J groupes de valeurs. Le cas standard peut être considéré comme un cas particulier, pour lequel $J = V$. Ici, V est supposé connu à l'avance alors que le nombre J de groupes est un paramètre à estimer.

Notations 2.8.

- N : nombre d'individus de l'échantillon
- Y : variable catégorielle à expliquer
- V : nombre de valeurs de la variable à expliquer (connu)
- J : nombre de groupes de valeurs de la variable à expliquer (inconnu)
- $j(v)$: index du groupe auquel est rattaché la valeur à expliquer v
- N_j : nombre d'individus du groupe à expliquer j
- m_j : nombre de valeurs à expliquer du groupe j

– n_v : nombre d'individus pour la valeur à expliquer v

Le nombre de groupes J étant fixé, on doit décrire une partition des valeurs à expliquer en J groupes, ce qui revient à spécifier $\{j(v)\}_{1\leq v\leq V}$. De façon similaire au cas du groupement des valeurs d'une variable explicative (formule 2.7), la spécification du groupement des valeurs à expliquer aboutit à l'ajout des nouveaux termes d'a priori suivants :

$$\log V + \log B(V, J). \tag{2.20}$$

Notons qu'une fois cette partition spécifiée, les nombres m_j de valeurs par groupe s'en déduisent et ne font donc par partie du paramétrage de modélisation.

On se ramène ensuite au cas classique de la classification supervisée présenté en section 2.1. Les modèles en grille sont exploités pour définir dans chaque cellule explicative la distribution des individus sur les J groupes à expliquer. L'effectif par groupe N_j pour l'ensemble de tous les individus est déduit par sommation sur des effectifs par groupe pour chaque cellule.

Chaque individu étant associé à un groupe à expliquer, il s'agit désormais de préciser à quelle valeur à expliquer il est associé. Pour ce faire, on décrit localement à chaque groupe j la distribution des individus du groupe sur les valeurs du groupe, au moyen d'un modèle multinômial de distribution des N_j individus du groupe sur ses m_j valeurs. Comme précédemment, on utilise un a priori uniforme pour le paramétrage de ce modèle multinômial, ce qui conduit pour chaque groupe à l'ajout du nouveau terme d'a priori suivant :

$$\log \binom{N_j + m_j - 1}{m_j - 1} \tag{2.21}$$

La vraisemblance de la distribution des individus sur les groupes est gérée par le modèle de classification standard. Il faut ici ajouter un terme de vraisemblance localement à chaque groupe pour la distribution des individus du groupe sur les valeurs à expliquer du groupe, au moyen d'un terme du multinôme :

$$\log N_j! - \sum_{\{v;j(v)=j\}} \log n_v! \tag{2.22}$$

En sommant sur l'ensemble des groupes à expliquer, on obtient

$$\log V + \log B(V, J) + \sum_{j=1}^{J} \log \binom{N_j + m_j - 1}{m_j - 1} \tag{2.23}$$

pour les termes d'a priori et

$$\sum_{j=1}^{J} \log N_j! - \sum_{v=1}^{V} \log n_v! \tag{2.24}$$

pour les termes de vraisemblance.

2.3.3 Grille multivariée

On synthétise ici l'évaluation d'un modèle de classification supervisée généralisée pour un nombre quelconque de variables explicatives, en se contentant de présenter les notations et la formule d'évaluation des modèles. Il s'agit essentiellement de la formule 2.13 issue du cas de la classification supervisée standard, enrichie des termes d'a priori 2.23 et de vraisemblance 2.24 relatifs à la modélisation du groupement des valeurs de la variable à expliquer.

Notations 2.9.

- N : nombre d'individus de l'échantillon
- Y : variable catégorielle à expliquer
- V : nombre de valeurs de la variable à expliquer (connu)
- K : nombre de variables explicatives
- $\mathbb{K} = \{X_1, X_2, \ldots, X_K\}$: ensemble des variables explicatives
- $\mathbb{K}_1$: sous-ensemble des variables explicatives de type numérique
- $\mathbb{K}_2$: sous-ensemble des variables explicatives de type catégoriel
- K_s : nombre de variables explicatives sélectionnées (inconnu)
- $\mathbb{K}_s$: sous-ensemble des variables explicatives sélectionnées ($|\mathbb{K}_s| = K_s$)
- $V_k, k \in \mathbb{K}_2$: nombre de valeurs de la variable catégorielle X_k
- I_k : taille de la partition univariée (en intervalles ou groupes) de la variable X_k (inconnu)
- $N_{i_1 i_2 \ldots i_K}$: nombre d'individus de la cellule $(i_1, i_2, \ldots, i_K)$
- $N_{i_1 i_2 \ldots i_K j}$: nombre d'individus de la cellule $(i_1, i_2, \ldots, i_K)$ pour la valeur à expliquer j
- J : nombre groupes de valeurs de la variable à expliquer (inconnu)
- $j(v)$: index du groupe auquel est rattaché la valeur à expliquer v
- N_j : nombre d'individus du groupe à expliquer j
- m_j : nombre de valeurs à expliquer du groupe j
- n_v : nombre d'individus pour la valeur à expliquer v

Théorème 2.7. *Un modèle de classification généralisée par grille suivant un a priori hiérarchique est optimal au sens de Bayes si son évaluation par la formule suivante est*

minimale sur l'ensemble de tous les modèles :

$$
\begin{aligned}
&\log V + \log B(V,J) + \sum_{j=1}^{J} \log \binom{N_j + m_j - 1}{m_j - 1} \\
&\log(K+1) + \log \binom{K + K_s - 1}{K_s} \\
&+ \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_1} \log N + \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_2} \log V_k \\
&+ \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_1} \log \binom{N + I_k - 1}{I_k - 1} + \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_2} \log B(V_k, I_k) \\
&+ \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_K=1}^{I_K} \log \binom{N_{i_1 i_2 \dots i_K} + J - 1}{J - 1} \\
&+ \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_K=1}^{I_K} \left(\log N_{i_1 i_2 \dots i_K}! - \sum_{j=1}^{J} \log N_{i_1 i_2 \dots i_K j}! \right) \\
&+ \sum_{j=1}^{J} \log N_j! - \sum_{v=1}^{V} \log n_v!
\end{aligned}
\tag{2.25}
$$

2.3.4 Interprétation

Dans le cas où aucune variable n'est sélectionnée et où les valeurs à expliquer sont partitionnées en un seul groupe ($K = 0$ et $J = 1$), la formule 2.25 se réduit à

$$
\log V + \log(K+1) + \log \binom{N + V - 1}{V - 1} + \log \frac{N!}{n_1! n_2! \dots n_V!}, \tag{2.26}
$$

ce qui correspond comme dans 2.14 à la probabilité a posteriori d'un modèle multinômial des valeurs à expliquer, indépendant des variables explicatives. La seule différence avec la formule 2.14 est l'ajout du terme $\log V$, qui correspond au choix du nombre de groupes pour la variable à expliquer.

Dans le cas où par contre les valeurs à expliquer sont partitionnées en autant de groupes que de valeurs ($J = V$), la formule 2.25 se réduit à

$$
\log V + \log B(V,V) + \log(K+1) + \log \binom{N + J - 1}{J - 1} + \log \frac{N!}{N_1! N_2! \dots n_J!}. \tag{2.27}
$$

Compte tenu du fait que les effectifs par groupes N_j sont égaux aux effectifs par valeur, on retrouve la formule précédente 2.26, avec un terme supplémentaire $\log B(V,V)$ correspondant à la description du groupement des valeurs à expliquer.

Il est intéressant de noter que les deux cas extrêmes du groupement de la variable à expliquer en un seul groupe ou en V groupes conduisent à des formules d'évaluation

quasiment identiques, alors que les termes de la formule 2.25 s'annulant ou non sont différents dans les deux cas.

Les cas intermédiaires ($1 < J < V$) offrent la potentialité de découverte de corrélation entre les variables explicatives et la variable à expliquer, dans le cas de nombreuses valeurs à expliquer.

2.4 Estimation de densité jointe

Dans ce qui précède, on a proposé des modèles d'estimation de densité conditionnelle $P(Y|X)$ dans tous les cas de figure pour des variables explicative ou à expliquer de type numérique ou catégorielle. On propose ici une adaptation de ces modèles au cas de l'estimation de densité jointe.

2.4.1 Discrétisation bivariée

Après avoir formalisé l'évaluation d'une grille dans le cas de deux variables numériques à expliquer, on montre que ce type de grille peut s'interpréter comme un modèle non paramétrique de corrélation entre les rangs de chaque variable.

2.4.1.1 Formalisation

Dans le cas de l'estimation de densité conditionnelle $P(Y|X)$, on connaît tout de la variable explicative X et on cherche à décrire la variable à expliquer Y connaissant X. Dans le cas de la densité jointe, on cherche à décrire conjointement les deux variables, qui auront toutes les deux le rôle de variable à expliquer. A cet effet, on les appellera Y_1 et Y_2.

On propose ici une famille de modèles où chaque variable à expliquer est partitionnée en intervalles. Dans le cas de la régression, on a procédé de façon hiérarchique, en distribuant les individus en premier lieu sur les intervalles de la variable explicative, en second lieu, par intervalle explicatif, sur les intervalles à expliquer. Ici, on distribue les individus sur l'ensemble des cellules de la grille bidimensionnelle. Cette distribution étant spécifiée, on en déduit par sommation sur les cellules les effectifs des intervalles pour chaque variable à expliquer. Il suffit alors de spécifier le rang des individus localement à chaque intervalle (pour chaque variable), pour connaître leur rang global. On a ainsi un modèle permettant de spécifier conjointement les rangs des deux variables.

Définition 2.13. Un modèle d'estimation de densité par grille est défini par :
- un nombre d'intervalles pour chaque variable à expliquer,
- la distribution des individus sur les cellules de la grille de données ainsi définie.

Notations 2.10.
- N : nombre d'individus de l'échantillon
- J_1, J_2 : nombre d'intervalles pour chaque variable (inconnu)
- $G = J_1 J_2$: nombre de cellules de la grille du modèle
- $N_{j_1}^{(1)}$: nombre d'individus de l'intervalle j_1 de la variable Y_1

– $N_{j_2}^{(2)}$: nombre d'individus de l'intervalle j_2 de la variable Y_2
– $N_{j_1 j_2}$: nombre d'individus de la cellule (j_1, j_2) de la grille

Un modèle d'estimation de densité par grille est entièrement caractérisé par le choix des paramètres $J_1, J_2, \{N_{j_1 j_2}\}_{1 \leq j_1 \leq J_1, 1 \leq j_2 \leq J_2}$. Les effectifs des intervalles $N_{j_1}^{(1)}$ et $N_{j_2}^{(2)}$ sont déduits par comptage des effectifs par cellule de la grille.

Définition 2.14. On appelle a priori hiérarchique l'a priori de modèle de densité par grille basé sur les hypothèses suivantes :
– les nombres d'intervalles J_1 et J_2 des variables à expliquer sont indépendants entre eux, et compris entre 1 et N de façon équiprobable,
– pour une grille de taille donnée (J_1, J_2), toutes les distributions des individus sur les G cellules de la grille sont équiprobables,
– pour un intervalle donné d'une variable à expliquer donnée, toutes les distributions des rangs des individus sont équiprobables.

Comme précédemment, l'a priori sur les paramètres des modèles est hiérarchique, uniforme à chaque étage de la hiérarchie. En utilisant la définition formelle des modèles et leur distribution a priori hiérarchique, la formule de Bayes permet de calculer de manière exacte la probabilité d'un modèle connaissant les données, ce qui conduit au théorème 2.8.

Théorème 2.8. *Un modèle d'estimation de densité par grille suivant un a priori hiérarchique est optimal au sens de Bayes si son évaluation par la formule suivante est minimale sur l'ensemble de tous les modèles :*

$$\begin{aligned} & 2 \log N + \log \binom{N+G-1}{G-1} \\ & + \log N! - \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \log N_{j_1 j_2}! \\ & + \sum_{j_1=1}^{J_1} \log N_{j_1}^{(1)}! + \sum_{j_2=1}^{J_2} \log N_{j_2}^{(2)}! \end{aligned} \tag{2.28}$$

La première ligne de la formule 2.28 regroupe des termes d'a priori correspondant au choix des nombres d'intervalles J_1 et J_2 et à la spécification de la distribution multinômiale des N individus de l'échantillon sur les G cellules de la grille. La deuxième ligne représente la vraisemblance de la distribution des individus dans les cellules de la grille, au moyen d'un terme du multinôme. La dernière ligne correspond à la vraisemblance des rangs localement à chaque intervalle pour chacune des variables à expliquer.

2.4.1.2 Interprétation

Dans le cas d'une grille de données comportant une seule cellule, la formule 2.28 se réduit à

$$2 \log N + 2 \log N!, \tag{2.29}$$

ce qui correspond essentiellement à $N!$ possibilités pour spécifier un rangement de N individus pour chacune des deux variables à expliquer. On retrouve deux fois la valeur du critère 2.16 correspondant à la régression d'une variable numérique en l'absence de variable explicative.

On peut alors interpréter le modèle d'estimation de densité jointe de la définition 2.13 comme un modèle de description de la corrélation entre les deux variables à expliquer. En cas d'indépendance entre les variables, la description des deux variables conjointement se réduit à la somme des descriptions de chaque variable individuellement. Le modèle en grille permet de capturer de façon non paramétrique des corrélations entre les rangs des variables à expliquer. Le surcoût de description du modèle de corrélation en grille est alors compensé par une description plus concise des rangs de chaque variable connaissant le modèle de corrélation. Le meilleur compromis est recherché suivant une approche Bayesienne de la sélection de modèles.

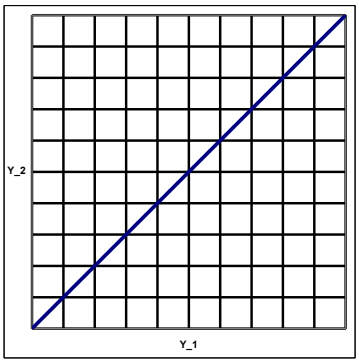


FIG. 2.8 – Grille de discrétisation bivariée avec 10 intervalles équidistribués pour deux variables à expliquer identiques $Y_1 = Y_2$.

Exemple de deux variables numériques à expliquer corrélées. Prenons l'exemple de deux variables identiques $Y_1 = Y_2$, et d'un modèle en grille M_J comportant J ($J = J_1 = J_2$) intervalles équidistribués, comme illustré sur la figure 2.8. Pour chaque variable à expliquer, chaque intervalle a un effectif N/J. Parmi les $G = J^2$ cellules de la grille, les J cellules sur la diagonale sont d'effectif N/J, les autres cellules étant vides. Le coût de description $c(M_J)$ de la grille est alors égal à :

$$c(M_J) = 2\log N + \log \binom{N + J^2 - 1}{J^2 - 1} + \log N! - J\log(N/J)! + 2J\log(N/J)! \qquad (2.30)$$

En fixant J et en utilisant l'approximation asymptotique $\log N! = N(\log N - 1)$ basée sur la formule de Stirling, on obtient $c(M_1) \approx 2N \log N$ et

$$c(M_J) \approx c(M_1) + (J^2 - 1)\log N - N \log J. \qquad (2.31)$$

La formule 2.31 montrent que dans le cas envisagé de corrélation parfaite entre deux variables à expliquer, le modèle en grille a un coût de description asymptotique décroissant avec la taille de la grille. Une expérimentation numérique dans le cas non asymptotique montre que la grille de coût minimal est obtenue pour $J \approx \sqrt{N}$.

2.4.2 Groupement de valeurs bivarié

Après avoir formalisé l'évaluation d'une grille dans le cas de deux variables catégorielles à expliquer, on montre que ce type de grille peut s'interpréter comme un modèle non paramétrique de corrélation entre les valeurs de chaque variable.

2.4.2.1 Formalisation

On cherche à décrire conjointement les valeurs des deux variables catégorielles à expliquer Y_1 et Y_2, comme illustré sur la figure 2.9.

D	∅	•	∅	•
C	•	∅	•	∅
B	∅	•	∅	•
A	•	∅	•	∅
	a	b	c	d

{B, D}	∅	●
{A, C}	●	∅
	{a, c}	{b, d}

FIG. 2.9 – Exemple de densité jointe pour deux variables catégorielles Y_1 ayant 4 valeurs a, b, c, d et Y_2 ayant 4 valeurs A, B, C, D. Le tableau de contingence sur la gauche ne contient des individus que sur la moitié des cases (marquées •), les autres cases étant vides. Suite au groupement de valeurs bivarié, le tableau de contingence sur la droite permet une description synthétique de la corrélation entre Y_1 et Y_2.

De façon analogue au cas de l'estimation de densité jointe pour deux variables à expliquer numériques, on propose une famille de modèles où chaque variable à expliquer est partitionnée, cette fois ci en groupes de valeurs. On distribue les individus sur l'ensemble des cellules de la grille bidimensionnelle ainsi définie. Cette distribution étant spécifiée, on en déduit par sommation sur les cellules les effectifs des groupes pour chaque variable à expliquer. On utilise alors le procédé décrit en 2.3.2 dans le cas du groupement des valeurs d'une variable à expliquer pour associer chaque individu à sa valeur à expliquer localement à son groupe, ceci pour chaque variable à expliquer.

Définition 2.15. Un modèle d'estimation de densité par grille est défini par :
- un nombre de groupes pour chaque variable à expliquer,
- la partition de chaque variable à expliquer en groupes de valeurs,
- la distribution des individus sur les cellules de la grille de données ainsi définie,
- la distribution des individus de chaque groupe sur les valeurs du groupe, pour chaque variable à expliquer.

Notations 2.11.
- N : nombre d'individus de l'échantillon
- V_1, V_2 : nombre de valeurs pour chaque variable (connu)
- J_1, J_2 : nombre de groupes pour chaque variable (inconnu)
- $G = J_1 J_2$: nombre de cellules de la grille du modèle
- $j^{(1)}(v_1), j^{(2)}(v_2)$: index du groupe auquel est rattachée la valeur v_1 (resp. v_2)
- $m_{j_1}^{(1)}, m_{j_2}^{(2)}$: nombre de valeurs du groupe j_1 (resp. j_2)
- $n_{v_1}^{(1)}, n_{v_2}^{(2)}$: nombre d'individus pour la valeur v_1 (resp. v_2)

– $N_{j_1}^{(1)}, N_{j_2}^{(2)}$: nombre d'individus du groupe j_1 (resp. j_2)
– $N_{j_1 j_2}$: nombre d'individus de la cellule (j_1, j_2) de la grille

Un modèle d'estimation de densité par grille est entièrement caractérisé par le choix des paramètres de partition des valeurs en groupes

$$J_1, J_2, \{j^{(1)}(v_1)\}_{1 \leq v_1 \leq V_1}, \{j^{(2)}(v_2)\}_{1 \leq v_2 \leq V_2},$$

des paramètres de distribution des individus sur les cellules de la grille

$$\{N_{j_1 j_2}\}_{1 \leq j_1 \leq J_1, 1 \leq j_2 \leq J_2},$$

et des paramètres de distribution des individus des groupes sur les valeurs des variables

$$\{n_{v_1}^{(1)}\}_{1 \leq v_1 \leq V_1}, \{n_{v_2}^{(2)}\}_{1 \leq v_2 \leq V_2}.$$

Les nombres de valeurs par groupe sont déduits du choix des partitions des valeurs en groupes. Les effectifs des groupes sont déduits par comptage des effectifs des cellules de la grille.

Définition 2.16. On appelle a priori hiérarchique l'a priori de modèle de densité par grille basé sur les hypothèses suivantes :
– les nombres de groupes de valeurs J_1 (resp. J_2) des variables à expliquer sont indépendants entre eux, et compris entre 1 et V_1 (resp. V_2) de façon équiprobable,
– pour un nombre de groupes donné J_1 de Y_1, toutes les partitions des V_1 valeurs en J_1 groupes sont équiprobables,
– pour un nombre de groupes donné J_2 de Y_2, toutes les partitions des V_2 valeurs en J_2 groupes sont équiprobables,
– pour une grille de taille donnée (J_1, J_2), toutes les distributions des N individus sur les G cellules de la grille sont équiprobables,
– pour un groupe donné d'une variable à expliquer donnée, toutes les distributions des individus sur les valeurs du groupe sont équiprobables.

Comme précédemment, l'a priori sur les paramètres des modèles est hiérarchique, uniforme à chaque étage de la hiérarchie. En utilisant la définition formelle des modèles et leur distribution a priori hiérarchique, la formule de Bayes permet de calculer de manière exacte la probabilité d'un modèle connaissant les données, ce qui conduit au théorème 2.9.

Théorème 2.9. *Un modèle d'estimation de densité par grille suivant un a priori hiérarchique est optimal au sens de Bayes si son évaluation par la formule suivante est minimale sur l'ensemble de tous les modèles :*

$$\begin{aligned}
&\log V_1 + \log V_2 + \log B(V_1, J_1) + \log B(V_2, J_2) \\
&+ \log \binom{N+G-1}{G-1} + \sum_{j_1=1}^{J_1} \log \binom{N_{j_1}^{(1)} + m_{j_1}^{(1)} - 1}{m_{j_1}^{(1)} - 1} + \sum_{j_2=1}^{J_2} \log \binom{N_{j_2}^{(2)} + m_{j_2}^{(2)} - 1}{m_{j_2}^{(2)} - 1} \\
&+ \log N! - \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \log N_{j_1 j_2}! \\
&+ \sum_{j_1=1}^{J_1} \log N_{j_1}^{(1)}! + \sum_{j_2=1}^{J_2} \log N_{j_2}^{(2)}! - \sum_{v_1=1}^{V_1} \log n_{v_1}^{(1)}! - \sum_{v_2=1}^{V_2} \log n_{v_2}^{(2)}!
\end{aligned} \tag{2.32}$$

La première ligne de la formule 2.32 regroupe des termes d'a priori correspondant au choix des nombres de groupes J_1 et J_2 et à la spécification de la partition de chaque variable à expliquer en groupes de valeurs. La deuxième ligne représente la spécification de la distribution multinômiale des N individus de l'échantillon sur les G cellules de la grille, suivi de la spécification de la distribution des individus de chaque groupe sur les valeurs du groupe. La troisième ligne représente la vraisemblance de la distribution des individus dans les cellules de la grille, au moyen d'un terme du multinôme. La dernière ligne correspond à la vraisemblance des valeurs localement à chaque groupe pour chacune des variables à expliquer.

2.4.2.2 Interprétation

Dans le cas d'une grille de données comportant une seule cellule, la formule 2.32 se réduit à :

$$\begin{aligned} &\log V_1 + \log V_2 + \log \binom{N+V_1-1}{V_1-1} + \log \binom{N+V_2-1}{V_2-1} \\ &+ \log \frac{N!}{n_{v_1}^{(1)}! n_{v_2}^{(1)}! \dots n_{V_1}^{(1)}!} + \log \frac{N!}{n_{v_1}^{(2)}! n_{v_2}^{(2)}! \dots n_{V_2}^{(2)}!} \end{aligned} \tag{2.33}$$

ce qui correspond à la probabilité a posteriori d'un modèle multinômial, pour chacune des variables catégorielles Y_1 et Y_2 à expliquer. On retrouve deux fois les termes du critère 2.26 correspondant à la classification généralisée d'une variable catégorielle en l'absence de variable explicative.

On peut alors interpréter le modèle d'estimation de densité jointe de la définition 2.15 comme un modèle de description de la corrélation entre les deux variables à expliquer. En cas d'indépendance entre les variables, la description des deux variables conjointement se réduit à la somme des descriptions de chaque variable individuellement. Le modèle en grille permet de capturer de façon non paramétrique des corrélations entre les valeurs des variables à expliquer. Le surcoût de description du modèle de corrélation en grille est alors compensé par une description plus concise des valeurs de chaque variable connaissant le modèle de corrélation. Le meilleur compromis est recherché suivant une approche Bayesienne de la sélection de modèles.

Exemple de deux variables catégorielles à expliquer corrélées. Prenons l'exemple de deux variables catégorielles identiques, et d'un modèle en grille M comportant autant de groupes que de valeurs ($J_1 = V_1$), comme illustré sur la figure 2.10. Le coût de description $c(M)$ de la grille est alors égal à :

$$2\log V_1 + 2\log B(V_1, V_1) + \log \binom{N+V_1^2-1}{V_1^2-1} + \log \frac{N!}{n_{v_1}^{(1)}! n_{v_2}^{(1)}! \dots n_{V_1}^{(1)}!} \tag{2.34}$$

En comparant les formules 2.33 dans le cas d'indépendance et 2.34 dans le cas d'égalité des variables à expliquer, on observe un surcoût de modélisation dans les termes d'a priori (spécification de chaque groupement de valeurs et spécification de la distribution des individus sur la grille bidimensionnelle). En revanche, le coût de vraisemblance est

d	$\emptyset$	$\emptyset$	$\emptyset$	$\bullet$
c	$\emptyset$	$\emptyset$	$\bullet$	$\emptyset$
b	$\emptyset$	$\bullet$	$\emptyset$	$\emptyset$
a	$\bullet$	$\emptyset$	$\emptyset$	$\emptyset$
	a	b	c	d

FIG. 2.10 – Grille de groupement de valeurs bivarié avec autant de groupes que de valeurs pour deux variables à expliquer identiques $Y_1 = Y_2$.

divisé par deux : les corrélations étant capturées dans le modèle en grille, la description des deux variables à expliquer se réduit à la description d'une seule variable.

Pour fixer les idées, comparons les formules 2.33 et 2.34 dans un cadre asymptotique. Le terme du multinôme d'une variable catégorielle peut être approximé de la façon suivante

$$\log \frac{N!}{n_{v_1}^{(1)}! n_{v_2}^{(1)}! \dots n_{V_1}^{(1)}!} \approx N H(Y_1),$$

où $H(Y_1)$ est l'entropie de Shannon de la variable Y_1 [Shannon, 1948]. L'évaluation du modèle d'estimation de densité jointe dans le cas d'une grille à une seule cellule (formule 2.33) est asymptotiquement égale à

$$2(V_1 - 1) \log N + 2N H(Y_1). \tag{2.35}$$

Dans le cas d'un modèle en grille ayant autant de groupes que de valeurs, l'évaluation (formule 2.34) est asymptotiquement égale à

$$(V_1^2 - 1) \log N + N H(Y_1). \tag{2.36}$$

Le modèle capturant la corrélation sera donc préféré au modèle d'indépendance, dès que le nombre d'individus sera suffisant par rapport au nombre de valeurs.

Il est à noter que les formules 2.33 et 2.34 permettent de choisir le meilleur modèle dans un cadre non asymptotique.

2.4.3 Grille multivariée

On synthétise ici l'évaluation d'un modèle d'estimation de densité jointe pour un nombre quelconque de variables à expliquer, en se contentant de présenter les notations et la formule d'évaluation des modèles. Comme dans les cas de la classification supervisée et de la régression, on incorpore un aspect sélection de variables pour le passage du bivarié au multivarié.

Notations 2.12.

- N : nombre d'individus de l'échantillon
- K : nombre de variables à expliquer
- $\mathbb{K} = \{Y_1, Y_2, \dots, Y_K\}$: ensemble des variables à expliquer

- $\mathbb{K}_1$: sous-ensemble des variables à expliquer de type numérique
- $\mathbb{K}_2$: sous-ensemble des variables à expliquer de type catégoriel
- K_s : nombre de variables à expliquer sélectionnées (inconnu)
- $\mathbb{K}_s$: sous-ensemble des variables à expliquer sélectionnées ($|\mathbb{K}_s| = K_s$)
- $V_k, k \in \mathbb{K}_2$: nombre de valeurs de la variable catégorielle X_k
- J_k : taille de la partition univariée (en intervalles ou groupes) de la variable Y_k (inconnu)
- $G = \prod_{k=1}^{K} J_k$: nombre de cellules de la grille du modèle
- $m_{j_k}^{(k)}, k \in \mathbb{K}_2$: nombre de valeurs du groupe j_k de la variable catégorielle Y_k
- $n_{v_k}^{(k)}, k \in \mathbb{K}_2$: nombre d'individus pour la valeur v_k de la variable catégorielle Y_k
- $N_{j_k}^{(k)}, k \in \mathbb{K}$: nombre d'individus de l'intervalle (ou du groupe de valeurs) j_k de la variable Y_k
- $N_{j_1 j_2 \dots j_K}$: nombre d'individus de la cellule $(j_1, i_2, \dots, j_K)$ de la grille

Théorème 2.10. *Un modèle d'estimation de densité jointe par grille suivant un a priori hiérarchique est optimal au sens de Bayes si son évaluation par la formule suivante est minimale sur l'ensemble de tous les modèles :*

$$\begin{aligned}
&\log(K+1) + \log \binom{K + K_s - 1}{K_s} \\
&+ \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_1} \log N + \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_2} \log V_k + \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_2} \log B(V_k, J_k) \\
&+ \log \binom{N + G - 1}{G - 1} + \sum_{k \in \mathbb{K}_2} \sum_{j_k=1}^{J_k} \log \binom{N_{j_k}^{(k)} + m_{j_k}^{(k)} - 1}{m_{j_k}^{(k)} - 1} \\
&+ \log N! - \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \dots \sum_{j_K=1}^{J_K} \log N_{j_1 j_2 \dots j_K}! \\
&+ \sum_{k \in \mathbb{K}} \sum_{j_k=1}^{J_k} \log N_{j_k}^{(k)}! - \sum_{k \in \mathbb{K}_2} \sum_{v_k=1}^{V_k} \log n_{v_k}^{(k)}!
\end{aligned} \tag{2.37}$$

2.4.4 Interprétation

Dans le cas où aucune variable n'est sélectionnée, la grille ne contient qu'une seule cellule et la formule 2.37 se réduit à

$$\log(K+1) + \sum_{k \in \mathbb{K}_2} \log \binom{N + V_k - 1}{V_k - 1} + \sum_{k \in \mathbb{K}_1} \log N! + \sum_{k \in \mathbb{K}_2} \log \frac{N!}{n_{v_k}^{(k)}! n_{v_k}^{(k)}! \dots n_{V_k}^{(k)}!} \tag{2.38}$$

ce qui correspond à la spécification des rangs de chaque variable numérique à expliquer et à la spécification au moyen d'un modèle multinômial des valeurs de chaque variable catégorielle à expliquer .

La taille $G = \prod_{k=1}^{K} J_k$ d'une grille multivariée croît très vite avec le nombre de variables à expliquer. Par exemple, pour un nombre de variables sélectionnées $K_s \approx \log_2 N$, la grille contient environ N cellules pour un effectif moyen d'un individu par cellule. Le

coût de description de la corrélation inter-variables devient alors un facteur limitant. La sélection de variables détermine un sous-ensemble de variables fortement corrélées, dont la description jointe est rendue plus concise. Les variables non sélectionnées sont décrites de façon indépendante comme dans le cas de la formule 2.38.

2.5 Généralisation

Dans ce qui précède, on a proposé des modèles d'estimation de densité pour une variable à expliquer catégorielle ou numérique conditionnellement à une ou plusieurs variables explicatives. On a également introduit des modèles d'estimation de densité pour deux à plusieurs variables à expliquer conjointement. On clôt ici la généralisation de l'approche en proposant des modèles d'estimation de densité pour un nombre quelconque de variables à expliquer, conditionnellement à un nombre quelconque de variables explicatives.

On introduit cette généralisation en prenant le cas de trois variables numériques, considérées successivement comme explicatives ou à expliquer. L'approche est ensuite généralisée à un nombre quelconque de variables, numériques ou catégorielles, explicatives ou à expliquer, en intégrant un aspect sélection de variables tant du côté explicatif que du côté à expliquer.

2.5.1 Cas de trois variables numériques

On applique dans cette section l'approche d'estimation de densité par grille pour un triplet de variables numériques dans trois cas de figure, selon le nombre de variables à expliquer considérées. Pour des raison de simplification, cette section n'intègre pas l'aspect sélection de variables dans les modèles considérés.

2.5.1.1 Densité conditionnelle $P(Y_3|X_1, X_2)$

Le modèle de régression $P(Y_3|X_1, X_2)$ a déjà été traité dans la section 2.2.

Les deux premières variables sont explicatives, la dernière étant à expliquer. On connaît tout des variables explicatives X_1 et X_2 , et on s'attache à décrire le rang des individus sur la variable à expliquer Y_3, connaissant leur rang sur les variables explicatives X_1 et X_2. A cet effet, on choisit indépendamment les bornes des intervalles pour chaque variable explicative. Les valeurs explicatives étant entièrement connues, cela détermine l'effectif de chaque cellule de la grille explicative bidimensionnelle. Il reste alors à spécifier dans chaque cellule explicative la distribution des individus sur les intervalles à expliquer. L'effectif global de chaque intervalle à expliquer se déduit par sommation des effectifs des cellules de la grille explicative. La description des rangs se fait localement à chaque intervalle à expliquer, ce qui définit le rang à expliquer global des individus.

Pour résumer, on exploite ici la nature (explicative ou à expliquer) des variables en hiérarchisant le choix des paramètres des modèles :

– choix du nombre d'intervalles par variable,
– choix des bornes des intervalles indépendamment pour chaque variable explicative,

– pour chaque cellule de la grille bidimensionnelle des intervalles explicatifs, choix de la distribution des individus sur les intervalles à expliquer.

Cette hiérarchisation des paramètres, avec un a priori uniforme à chaque étage, conduit au critère d'évaluation des modèles donné par la formule 2.39, en utilisant les notations habituelles.

$$\begin{aligned} &3\log N + \log\binom{N+I_1-1}{I_1-1} + \log\binom{N+I_2-1}{I_2-1} + \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\log\binom{N_{i_1i_2}+J_3-1}{J_3-1} \\ &+ \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2} N_{i_1i_2}! - \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\sum_{j_3=1}^{J_3}\log N_{i_1i_2j_3}! + \sum_{j_3=1}^{J_3}\log N^{(3)}_{j_3}! \end{aligned} \tag{2.39}$$

Les deux premiers termes de la deuxième ligne correspondent au terme de vraisemblance de la distribution multinômiale des individus de chaque cellule de la grille explicative sur les intervalles à expliquer.

Dans le cas d'une grille mono-cellule, l'évaluation se réduit à $3\log N + \log N!$, ce qui correspond à la description des rangs pour la variable à expliquer.

2.5.1.2 Densité conditionnelle $P(Y_2, Y_3|X_1)$

La première variable est explicative, les deux autres étant à expliquer. On connaît tout de la variable explicative X_1, et on s'attache à décrire le rang des individus conjointement sur les deux variables Y_2 et Y_3 à expliquer, connaissant leur rang sur la variable explicative X_1. Cette différence de statut entre les variables permet d'introduire une hiérarchie dans le choix des paramètres de modélisation par grille :

– choix du nombre d'intervalles par variable,
– choix des bornes des intervalles de la variable explicative,
– pour chaque intervalle explicatif, choix de la distribution des individus sur les cellules de la grille bidimensionnelle à expliquer.

Cette hiérarchisation des paramètres, avec un a priori uniforme à chaque étage, conduit au critère d'évaluation des modèles donné par la formule 2.40, en notant $G = J_2J_3$ la taille de la grille à expliquer.

$$\begin{aligned} &3\log N + \log\binom{N+I_1-1}{I_1-1} + \sum_{i_1=1}^{I_1}\log\binom{N^{(1)}_{i_1}+G-1}{G-1} \\ &+ \sum_{i_1=1}^{I_1}\log N^{(1)}_{i_1}! - \sum_{i_1=1}^{I_1}\sum_{j_2=1}^{J_2}\sum_{j_3=1}^{J_3}\log N_{i_1j_2j_3}! + \sum_{j_2=1}^{J_2}\log N^{(2)}_{j_2}! + \sum_{j_3=1}^{J_3}\log N^{(3)}_{j_3}! \end{aligned} \tag{2.40}$$

Les deux premiers termes de la deuxième ligne correspondent au terme de vraisemblance de la distribution multinômiale des individus de chaque intervalle explicatif sur les cellules de la grille bidimensionnelle à expliquer.

Dans le cas d'une grille mono-cellule, l'évaluation se réduit à $3\log N + 2\log N!$, ce qui correspond à la description des rangs pour chacune des deux variables à expliquer.

2.5.1.3 Densité jointe $P(Y_1, Y_2, Y_3)$

L'estimation de densité jointe $P(Y_1, Y_2, Y_3)$ a déjà été traitée dans la section 2.4.

Dans le cas de trois variables à expliquer, l'objectif d'un modèle d'estimation de densité jointe est de décrire conjointement le rang des individus sur les trois variables. L'utilisation d'une grille permet de décrire la distribution des individus sur les cellules de la grille. Pour chaque variable à expliquer, l'effectif par intervalle se déduit par sommation des effectifs des cellules de la grille sur les autres variables. La description des rangs se fait localement à chaque intervalle, ce qui définit le rang global des individus pour chaque variable. Le problème se ramène essentiellement au choix du nombre d'intervalles retenus pour chaque variable, ce qui détermine la résolution de la grille.

En utilisant les notations habituelles et en notant $G = J_1 J_2 J_3$ la taille de la grille à expliquer, le critère d'évaluation d'un modèle d'estimation de densité jointe est donné par la formule 2.41.

$$\begin{aligned} &3 \log N + \log \binom{N+G-1}{G-1} \\ &+ \log N! - \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \sum_{j_3=1}^{J_3} \log N_{j_1 j_2 j_3}! + \sum_{j_1=1}^{J_1} \log N_{j_1}^{(1)}! + \sum_{j_2=1}^{J_2} \log N_{j_2}^{(2)}! + \sum_{j_3=1}^{J_3} \log N_{j_3}^{(3)}! \end{aligned} \tag{2.41}$$

Les deux premiers termes de la deuxième ligne correspondent au terme de vraisemblance de la distribution multinômiale des individus de l'ensemble de l'échantillon sur les cellules de la grille tridimensionnelle à expliquer.

Dans le cas d'une grille mono-cellule, l'évaluation se réduit à $3 \log N + 3 \log N!$, ce qui correspond à la description des rangs pour chacune des trois variables à expliquer.

2.5.2 Grille multivariée

On synthétise ici l'évaluation d'un modèle d'estimation de densité conditionnelle pour un nombre quelconque de variables explicatives ou à expliquer, numériques ou catégorielles. Il s'agit de décrire pour chaque variable à expliquer le rang des individus dans le cas numérique et leur valeur dans le cas catégoriel, conditionnellement aux variables explicatives.

A cet effet, on utilise d'une part une grille explicative, d'autre part une grille à expliquer. L'objectif de la grille explicative est de partitionner les individus en cellules homogènes vis à vis de leur distribution sur les cellules de la grille à expliquer. La grille à expliquer permet de décrire plusieurs variables à expliquer conjointement, de façon plus concise qu'en les décrivant séparément.

Les dimensions de la grille explicative sont déterminées en partitionnant chaque variable explicative individuellement, en intervalles ou groupes de valeurs selon son type. Par contre, l'effectif des cellules n'est pas modélisé : ces effectifs sont déduits en distribuant les individus de l'échantillon dans les cellules explicatives en fonction des spécifications de partitionnement univarié.

Le procédé de modélisation est inverse pour la grille à expliquer. On modélise l'effectif de chaque cellule de la grille à expliquer, en décrivant pour chaque cellule explicative la distribution des individus de la cellule explicative sur les cellules à expliquer. Par sommation sur l'ensemble des cellules explicatives, on déduit les effectifs globaux des cellules à expliquer. Comme la grille à expliquer définit une partition factorisable sur les partition univariées à expliquer, les effectifs par intervalle ou groupe de valeurs des partitions univariées à expliquer sont déduit par sommation sur les cellules à expliquer.

Comme dans les cas multivariés précédents, on incorpore explicitement un aspect sélection de variables, tant pour la grille explicative que pour la grille à expliquer.

Notations 2.13.

- N : nombre d'individus de l'échantillon
- K : nombre total de variables
- K_X : nombre de variables explicatives
- K_Y : nombre de variables à expliquer
- $\mathbb{K}_X = \{X_1, X_2, \ldots, X_{K_X}\}$: ensemble des variables explicatives ($K_X = |\mathbb{K}_X|$)
- $\mathbb{K}_Y = \{Y_1, Y_2, \ldots, Y_{K_Y}\}$: ensemble des variables à expliquer ($K_Y = |\mathbb{K}_Y|$)
- $\mathbb{K} = \mathbb{K}_X \cup \mathbb{K}_Y$: ensemble de toutes les variables ($K = |\mathbb{K}|$)
- $\mathbb{K}_1$: sous-ensemble de $\mathbb{K}$ des variables numériques
- $\mathbb{K}_2$: sous-ensemble de $\mathbb{K}$ des variables catégorielles
- K_s : nombre de variables à expliquer sélectionnées (inconnu)
- $\mathbb{K}_s$: sous-ensemble de $\mathbb{K}$ des variables sélectionnées ($|\mathbb{K}_s| = K_s$)
- $V_k, k \in \mathbb{K}_2$: nombre de valeurs de la variable catégorielle X_k ou Y_k
- I_k, J_k : taille de la partition univariée (en intervalles ou groupes) de la variable X_k ou Y_k (inconnu)
- $G = \prod_{k \in \mathbb{K}_Y} J_k$: nombre de cellules de la grille à expliquer du modèle
- $m_{j_k}^{(k)}, k \in \mathbb{K}_2 \cap \mathbb{K}_Y$: nombre de valeurs du groupe j_k de la variable catégorielle à expliquer Y_k
- $n_{v_k}^{(k)}, k \in \mathbb{K}_2 \cap \mathbb{K}_Y$: nombre d'individus pour la valeur v_k de la variable catégorielle à expliquer Y_k
- $N_{j_k}^{(k)}, k \in \mathbb{K}_Y$: nombre d'individus de l'intervalle (ou du groupe de valeurs) j_k de la variable à expliquer Y_k
- $N_{i_1 i_2 \ldots i_{K_X}}$: nombre d'individus de la cellule $(i_1 i_2 \ldots i_{K_X})$ de la grille explicative
- $N_{i_1 i_2 \ldots i_{K_X} j_1 j_2 \ldots j_{K_Y}}$: nombre d'individus de la cellule $(j_1, j_2, \ldots, j_{K_Y})$ de la grille à expliquer pour la cellule $(i_1 i_2 \ldots i_{K_X})$ de la grille explicative

Théorème 2.11. *Un modèle de d'estimation de densité conditionnelle par grille suivant un a priori hiérarchique est optimal au sens de Bayes si son évaluation par la formule*

suivante est minimale sur l'ensemble de tous les modèles :

$$
\begin{aligned}
&\log(K+1) + \log \binom{K+Ks-1}{K_s} \\
&+ \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_1} \log N + \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_2} \log V_k \\
&+ \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_1 \cap \mathbb{K}_X} \log \binom{N+I_k-1}{I_k-1} + \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_2 \cap \mathbb{K}_X} \log B(V_k, I_k) + \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_2 \cap \mathbb{K}_Y} \log B(V_k, J_k) \\
&+ \sum_{k \in \mathbb{K}_2 \cap \mathbb{K}_Y} \sum_{j_k=1}^{J_k} \log \binom{N_{j_k}^{(k)} + m_{j_k}^{(k)} - 1}{m_{j_k}^{(k)} - 1} \\
&+ \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_{K_X}=1}^{I_{K_X}} \log \binom{N_{i_1 i_2 \ldots i_{K_X}} + G - 1}{G-1} \\
&+ \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_{K_X}=1}^{I_{K_X}} \left(\log N_{i_1 i_2 \ldots i_{K_X}}! - \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \cdots \sum_{j_{K_Y}=1}^{J_{K_Y}} \log N_{i_1 i_2 \ldots i_{K_X} j_1 j_2 \ldots j_{K_Y}}! \right) \\
&+ \sum_{k \in \mathbb{K}_Y} \sum_{j_k=1}^{J_k} \log N_{j_k}^{(k)}! - \sum_{k \in \mathbb{K}_2 \cap \mathbb{K}_Y} \sum_{v_k=1}^{V_k} \log n_{v_k}^{(k)}!
\end{aligned}
\tag{2.42}
$$

2.5.3 Interprétation

Dans le cas où aucune variable n'est sélectionnée, la grille ne contient qu'une seule cellule et la formule 2.42 se réduit à

$$
\begin{aligned}
&\log(K+1) + \sum_{k \in \mathbb{K}_2 \cap \mathbb{K}_Y} \log \binom{N+V_k-1}{V_k-1} \\
&+ \sum_{k \in \mathbb{K}_1 \cap \mathbb{K}_Y} \log N! + \sum_{k \in \mathbb{K}_2 \cap \mathbb{K}_Y} \log \frac{N!}{n_{v_k}^{(k)}! n_{v_k}^{(k)}! \ldots n_{V_k}^{(k)}!}
\end{aligned}
\tag{2.43}
$$

ce qui correspond comme dans le cas de l'estimation de densité jointe à la spécification des rangs de chaque variable numérique à expliquer et à la spécification au moyen d'un modèle multinômial des valeurs de chaque variable catégorielle à expliquer .

Dans un cadre plus général, la grille à expliquer permet de capturer les corrélations entre les variables à expliquer, ce qui permet de simplifier leur description jointe. La grille à explicative permet de capturer les corrélation entre les variables explicatives et les variables à expliquer.

Remarque 2.3. *Il est important de noter que toutes les formules de ce chapitre sont des cas particulier de la formule générale 2.42. Notamment, on retrouve :*

- *la classification supervisée dans le cas d'une seule variable explicative catégorielle, standard (section 2.1) ou généralisée (section 2.3) selon que le groupement des valeurs à expliquer soit interdit ou autorisé,*

- *la régression (section 2.2) dans le cas d'une seule variable explicative numérique,*
- *l'estimation de densité jointe (section 2.4) dans le cas d'au moins deux variables à expliquer, en l'absence de variable explicative.*

Remarque 2.4. *Habituellement, on distingue apprentissage supervisé et non supervisé selon qu'il y ait ou non une variable à expliquer. L'optique est ici inversée : le cas non supervisé correspond au cas où toutes les variables sont à expliquer.*

Remarque 2.5. *Alors que l'apprentissage non supervisé s'appuie généralement sur une métrique entre individus pour identifier des groupes d'individus homogènes, l'approche proposée est valide tant pour des variables numériques que catégorielles. La notion d'homogénéité entre individus est capturée dans les cellules de la grille, qui concentrent des groupes d'individus présentant de fortes corrélations entre les variables à expliquer.*

2.6 Conclusion

L'estimation de densité par grille se fait en distribuant les données sur une grille de cellules, dont les dimensions sont déterminées par un partitionnement de chaque variable. Ce partitionnement est obtenu par discrétisation pour les variables numériques et par groupement de valeurs pour les variables catégorielles. On suppose la densité constante dans chaque cellule de données de la grille ainsi définie.

L'enjeu principal est de régulariser cette approche, afin de trouver le bon niveau de résolution de la grille d'estimation de densité. La solution proposée se base sur une approche Bayesienne de la sélection de modèles. Toutes les probabilités sont estimées en statistiques discrètes, en utilisant la statistique de rang pour les variables numériques et la statistique multinômiale pour les variables catégorielles. Dans chaque situation, les informations disponibles (principalement dans les variables explicatives) sont utilisées au maximum pour définir un paramétrage adapté des modèles d'estimation de densité par grille. La structure "naturelle" des paramètres des modèles est exploitée afin de hiérarchiser la distribution a priori des modèles, en utilisant une distribution uniforme à chaque étape de la hiérarchie. Enfin, les distributions par cellule des grilles d'estimation de densité sont supposées indépendantes les unes des autres, ce qui favorise les modèles fortement discriminants. L'utilisation de cette démarche aboutit à une définition précise des paramètres des modèles et de leur distribution a priori. Il est alors possible de calculer exactement la probabilité d'un modèle connaissant les données, ce qui conduit à un critère d'évaluation des modèles optimal au sens de Bayes.

La famille des modèles en grille est très expressive et possède les qualités d'un approximateur universel. Les modèles en grille s'appliquent à tous les cas de l'estimation de densité, pour des variables numériques comme catégorielles, pour la densité jointe ou conditionnelle, en univarié, bivarié et multivarié.

3

Algorithmes d'optimisation des grilles de données

Sommaire

Dans le chapitre 2, nous avons introduit les modèles en grille pour l'estimation de densité conditionnelle ou jointe dans les tableaux de données individus * variables. Ces modèles sont évalués au moyen d'un critère analytique qui est optimal au sens de Bayes, mais difficile à optimiser. L'objectif de ce chapitre est de décrire des algorithmes efficaces pour l'optimisation des modèles en grille. Leur évaluation sur des jeux de données artificiels et réels est traitée dans le chapitre 4.

Les modèles en grille sont basés sur le partitionnement des N individus et K variables d'un tableau de données en une grille de cellules, résultant du produit cartésien de partitions univariées pour chaque variable. Les partitions univariées considérées sont les discrétisations dans le cas numérique et les groupements de valeurs dans le cas catégoriel.

Ces modèles sont très expressifs et leur nombre augmente de façon exponentielle avec la taille des données. Ainsi, pour une variable numérique, la taille de l'espace des modèles de discrétisation univariée est de l'ordre de $O(2^N)$, chaque discrétisation comportant $O(N)$ intervalles. Pour une variable catégorielle comportant V valeurs, le nombre de groupements de valeurs possible augmente plus rapidement encore, puisqu'il est basé sur le nombre de Bell. Pour un ensemble de K variable numériques, la taille de l'espace des modèles en grille est de l'ordre de $O((2^N)^K)$, chaque modèle comportant $O(N^K)$ cellules. Une évaluation exhaustive de l'ensemble des modèles de discrétisation multivariée aurait un coût de calcul de $O((2^N)^K N^K)$, ce qui est inenvisageable.

Nous présentons dans ce chapitre des méthodes d'optimisation des modèles en grille de complexité algorithmique $O(KN\sqrt{N}\log N \max(K, \log N))$ en temps de calcul dans le cas le plus général, avec une occupation mémoire $O(KN)$ égale à celle du tableau de données. Cette complexité algorithmique est obtenue essentiellement en exploitant une propriété d'additivité pour les critères d'évaluation des modèles en grille et la faible densité des individus dans les cellules d'une grille.

La section 3.1 introduit la notion d'additivité pour les critères d'évaluation d'une partition univariée et présente de façon synthétique les algorithmes d'optimisation associés. La section 3.2 étend cette notion d'addivité au cas multivarié des modèles de partitionnement en grille, et présente les principes permettant l'optimisation efficace de ces critères. La section 3.3 décrit l'application de ces algorithmes pour l'optimisation des modèles en grille dans le cas de la classification supervisée et de l'estimation de densité jointe, puis propose une extension de ces algorithmes dans le cas le plus général de l'estimation de densité conditionnelle pour un nombre quelconque de variables explicatives ou à expliquer. Enfin, la section 3.4 conclut ce chapitre.

Le rôle de ce chapitre est de présenter de façon synthétique les principes des algorithmes mis en oeuvre pour l'optimisation des modèles en grille. La description détaillée des algorithmes d'optimisation est reportée en annexe :

- l'annexe A reproduit l'article [Boullé, 2006b] sur la discrétisation supervisée,
- l'annexe B reproduit l'article [Boullé, 2005a] sur le groupement de valeurs supervisé,
- l'annexe D reproduit l'article [Boullé, 2007c] sur les algorithmes d'optimisation des modèles en grille dans le cas particulier du partitionnement bivarié.

3.1 Optimisation d'un critère additif de partitionnement univarié

Cette section introduit la notion d'additivité pour les critères d'évalution d'une partition univariée et présente les principes généraux d'optimisation de tels critères. Les cas particuliers de la discrétisation et du groupement de valeurs sont ensuite abordés.

3.1.1 Critère additif

Soit X une variable instanciée sur N individus, et M un modèle de partitionnement univarié de X en I parties $P_1, P_2, \ldots, P_I$. La définition 3.1 introduit la notion d'additivité pour les critères d'évaluation des modèles de partitionnement univarié.

Définition 3.1. Un critère d'évaluation $c(M)$ d'un modèle M de partitionnement univarié est *additif* s'il peut s'écrire sous la forme

$$c(M) = c^{(V)}(X, I) + \sum_{i=1}^{I} c^{(P)}(P_i) \tag{3.1}$$

où

- $c^{(V)}(X, I)$ ne dépend que des caractéristiques de la variable X et de la taille I de la partition,
- $c^{(P)}(P_i)$ ne dépend que des caractéristiques de la partie P_i.

L'intérêt principal de l'additivité d'un critère d'évaluation est de limiter l'impact d'une modification élémentaire d'une partition. Par exemple, lors de la fusion de deux parties adjacentes, l'impact sur le coût de partition se limite essentiellement aux deux parties concernées, les coûts des $I - 2$ autres parties restant inchangés.

3.1.2 Algorithme d'optimisation

L'algorithme d'optimisation proposé se compose d'une étape d'optimisation au moyen d'une heuristique gloutonne, suivie d'une étape de post-optimisation de la partition obtenue.

Heuristique gloutonne ascendante. L'algorithme 3.1 décrit une heuristique générique d'optimisation d'une partition univariée, classique dans la littérature dans le cas de la discrétisation supervisée [Zighed and Rakotomalala, 2000]. En partant d'une partition initiale comportant autant de parties élémentaires que d'individus, on évalue toutes les fusions possibles entre parties adjacentes. La meilleure fusion est effectuée si elle améliore le coût de la partition, et l'algorithme est réitéré tant qu'il y a amélioration du coût.

Post-optimisation de la taille de la partition. Afin de sortir d'un éventuel optimum local au moyen de plusieurs fusions successives, la procédure de fusion des parties de l'algorithme 3.1 est réitérée jusqu'à l'obtention d'une partition en une seule partie. La meilleure partition rencontrée au cours de ces fusions "forcées" est retournée en sortie.

Post-optimisation de la partition. L'algorithme 3.1 permet de parcourir efficacement $O(N)$ partitions de taille décroissante, mais, ne remettant jamais en question ses décisions de fusions de parties, il ne peut échapper aux optimaux locaux. La post-optimisation de la partition considère un voisinage local d'une partition basé sur des opérations élémentaires entre parties adjacentes : découpage d'une partie en deux, changement de frontières entre deux parties, fusion de deux parties... L'exploration systématique de ce voisinage permet

Algorithme 3.1 Heuristique d'optimisation gloutonne ascendante

Entrées: M {Partition initiale}
Sorties: $M^*, c(M^*) \leq c(M)$ {Partition finale optimisée}

```
M* ← M
Tant que amélioration faire
   {Recherche de la meilleure amélioration}
   M+ ← M*
   Pour tout Fusion m entre deux parties adjacentes faire
      {Evaluation de l'impact de la fusion m sur la partition M*}
      M' ← M* + m
      Si c(M') < c(M+) alors
         M+ ← M'
      Fin si
   Fin pour
   {Test d'amélioration}
   Si c(M+) < c(M*) alors
      M* ← M+
   Fin si
Fin tant que
```

d'échapper à une gamme importante d'optimaux locaux et d'améliorer la qualité de la partition obtenue.

3.1.3 Cas d'une variable numérique

Les algorithmes de discrétisation résumés dans cette section sont détaillés en annexe A (section 3).

Dans le cas de la discrétisation d'une variable numérique X instanciée sur N individus, les parties sont des intervalles, et deux parties sont adjacentes si les intervalles correspondants sont contigus. Il y a par conséquent $O(N)$ parties et $O(N)$ fusions de parties à considérer.

Algorithme optimal. Dans le cadre d'un critère additif, il est possible de trouver la discrétisation optimale en $O(N^3)$ au moyen d'un algorithme de programmation dynamique, décrit à plusieurs reprises dans la littérature [Fischer, 1958, Lechevallier, 1990, Fulton *et al.*, 1995, Elomaa and Rousu, 1996]. Cet algorithme est inutilisable en pratique sur les grandes bases de données, mais il est utile pour évaluer la qualité des heuristiques d'optimisation.

Heuristique gloutonne. L'heuristique gloutonne ascendante 3.1 nécessite $O(N^3)$ avec une implémentation naïve. En effet, à chacune des $O(N)$ étapes d'optimisation, $O(N)$ fusions de parties adjacentes sont évaluées, chaque évaluation étant basée sur un critère portant sur $O(N)$ parties par partition.

L'additivité du critère permet d'éviter les calculs inutiles en mémorisant les résultats intermédiaires et en réduisant l'impact de chaque fusion aux parties considérées. Couplé avec l'utilisation d'une liste triée maintenable de type AVL [Adelson-Velskii and Landis, 1962] pour la gestion des évaluations de fusions de parties, l'heuristique gloutonne peut être implémentée en $O(N \log N)$.

Post-optimisation de la taille de la partition. L'application exhaustive de l'heuristique gloutonne pour post-optimiser la taille de la partition est également implémentable en $O(N \log N)$.

Post-optimisation de la partition. Les voisinages envisagés pour une discrétisation sont basés sur des combinaisons de découpages ou de fusions d'intervalles adjacents :

- suppression d'un intervalle, par fusion de trois intervalles adjacents existants suivie d'un découpage de l'intervalle fusionné,
- ajustement de frontières entre deux intervalles, par fusion de deux intervalles adjacents existants suivie d'un découpage de l'intervalle fusionné,
- ajout d'un nouvel intervalle, par découpage d'un intervalle existant.

L'additivité du critère permet le parcours exhaustif du voisinage d'une discrétisation en $O(N)$, et la maintenance en $O(N \log N)$ de la liste triée des discrétisations voisines de la meilleure discrétisation courante.

3.1.4 Cas d'une variable catégorielle

Les algorithmes de groupement de valeurs résumés dans cette section sont détaillés en annexe B (section 2.4).

Dans le cas du groupement d'une variable catégorielle X instanciée sur N individus et comportant V valeurs, les parties sont des groupes de valeurs, et toutes les parties sont adjacentes deux à deux. Il y a par conséquent $O(V)$ parties et $O(V^2)$ fusions de parties à considérer. Si l'on veut garder un temps de calcul en $O(N \log N)$, cela demande une attention spéciale dès que V dépasse $\sqrt{N}$.

Algorithme optimal. Dans le cas le plus général du groupement de valeurs, il n'existe pas à notre connaissance dans la littérature d'algorithme optimal connu autre que l'algorithme exhaustif.

Heuristique gloutonne. L'heuristique gloutonne ascendante 3.1 nécessite $O(N \log N + V^4)$ avec une implémentation naïve. En effet, à chacune des $O(V)$ étapes d'optimisation, $O(V^2)$ fusions de parties adjacentes sont évaluées, chaque évaluation étant basée sur un critère portant sur $O(V)$ parties par partition. Le terme en $O(N \log N)$ dans la complexité correspond au calcul des effectifs par valeur à expliquer, par cumul sur les N individus triés par valeur.

L'additivité du critère permet d'éviter les calculs inutiles en mémorisant les résultats intermédiaires et en réduisant l'impact de chaque fusion aux parties considérées. Couplée

avec l'utilisation d'une liste triée maintenable, l'heuristique gloutonne peut être implémentée en $\mathrm{O}(N \log N + V^2 \log V)$.

En partant d'une solution comportant un nombre de groupes de valeurs $I \leq I_{Max} = \max(V, \sqrt{N})$, on se ramène à une complexité en $\mathrm{O}(N \log N)$.

Dans le cas du groupement supervisé de valeurs détaillé en annexe B, des algorithmes de prétraitement dédiés sont proposés pour initialiser efficacement une partition de qualité en I_{Max} groupes de valeurs.

Post-optimisation de la taille de la partition. L'application exhaustive de l'heuristique gloutonne pour post-optimiser la taille de la partition est également implémentable en $\mathrm{O}(N \log N)$.

Post-optimisation de la partition. Les voisinages envisagés pour un groupement de valeurs sont basés sur les déplacements de valeurs entre groupes de valeurs. Pour un groupement de V valeurs en I groupes, $\mathrm{O}(VI)$ déplacements de valeurs sont ainsi évalués.

La complexité $\mathrm{O}(VI)$ est en pratique du même ordre que $\mathrm{O}(N)$, puisque du fait de la forte régularisation des groupements de valeurs basée sur le nombre de Bell, le nombre de groupes I décroît très rapidement quand le nombre de valeurs V augmente.

3.2 Optimisation d'un critère additif de partitionnement multivarié

Cette section introduit la notion d'additivité pour les critères d'évaluation d'une partition multivariée relative à un modèle en grille. Les principes généraux d'optimisation de tels critères, sont ensuite présentés de façon macroscopique.

Les algorithmes correspondants sont détaillés en annexe D dans le cas particulier des partitions bivariées, notamment en ce qui concerne les preuves pour les complexités algorithmiques annoncées.

3.2.1 Critère additif

Soient $X_1, X_2, \ldots, X_K$ un ensemble de K variables instanciées sur N individus, et M un modèle de partitionnement multivarié en grille. Chaque variable X_k est partitionnée en I_k parties $P_1^{(k)}, P_2^{(k)}, \ldots, P_{I_k}^{(k)}$. Le produit cartésien des K partitions univariées définit une partition multivariée des individus en une grille de cellules $C_{i_1 i_2 \ldots i_K}$ de dimension $\mathcal{I} = (I_1, I_2, \ldots, I_K)$.

La définition 3.2 introduit la notion d'additivité pour les critères d'évaluation des modèles de partitionnement multivarié.

Définition 3.2. Un critère d'évaluation $c(M)$ d'un modèle M de partitionnement multivarié est *additif* s'il peut s'écrire sous la forme

$$\begin{aligned} c(M) = c^{(G)}\left(\mathcal{I}\right) + \sum_{k=1}^{K} c^{(V)}\left(X_k, I_k\right) \\ + \sum_{k=1}^{K}\sum_{i_k=1}^{I_k} c^{(P)}\left(P_{i_k}^{(k)}\right) + \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_k=1}^{I_K} c^{(C)}\left(C_{i_1 i_2 \ldots i_K}\right) \end{aligned} \tag{3.2}$$

où

- $c^{(G)}(\mathcal{I})$ ne dépend que des tailles $I_1, I_2, \ldots, I_K$ des partitions univariées de la grille,
- $c^{(V)}(X_k, I_k)$ ne dépend que des caractéristiques de la variable X_k et de la taille I_k de sa partition univariée,
- $c^{(P)}(P_{i_k}^{(k)})$ ne dépend que des caractéristiques de la partie $P_{i_k}^{(k)}$,
- $c^{(V)}(C_{i_1 i_2 \ldots i_K})$ ne dépend que des caractéristiques de la cellule $C_{i_1 i_2 \ldots i_K}$ et est nul pour les cellules vides.

3.2.2 Principes généraux d'optimisation

On décrit ci-dessous les principes généraux permettant l'implémentation efficace des algorithmes d'optimisation des modèles en grille avec critère d'évaluation additif. La mise en oeuvre effective de ces principes est relativement sophistiquée. Elle est détaillée dans le cas particulier du partitionnement bivarié en annexe D.

Utilisation de l'additivité du critère. L'additivité d'un critère d'évaluation permet de limiter l'impact d'une modification élémentaire d'une partition. Par exemple, lors de la fusion de deux parties adjacentes d'une variable, l'impact sur le coût de partition se limite essentiellement aux deux parties concernées et aux cellules non vides qu'elles contiennent.

Exploitation de la faible densité des individus dans la grille. Alors qu'une grille de K variables et N individus comporte potentiellement N^K cellules, le nombre de cellules non vides est majoré par N. Dans le cas multivarié, les grilles de données sont donc des structures fortement creuses. Cette faible densité des individus peut être exploitée au moyen de structure de données appropriées, pour ne traiter que les cellules non vides d'une grille.

Accès optimisé à une cellule par sa signature. Une grille multivariée est un produit cartésien de K partitions univariées. Chaque cellule $C_{i_1 i_2 \ldots i_K}$ d'une grille est identifiée par sa *signature* $(P_{i_1}^{(1)}, P_{i_2}^{(2)}, \ldots, P_{i_K}^{(K)})$ dans ce produit cartésien.

La signature d'une cellule sert de base au calcul d'une clé de hachage, permettant de tester en O(1) l'existence d'une cellule non vide au moyen d'une table de hachage [Knuth, 1997]. Le calcul de la clé de hachage demande $O(K)$ opérations, puisque la signature est de longueur K.

Utilisation de fonctions de hachage incrémentales. La signature d'une cellule est utilisée pour fabriquer un identifiant entier $Id(C_{i_1 i_2 \ldots i_K}) = \sum_{k=0}^{K-1} Id(P_{i_k}^{(k)}) N^k$ compris entre 0 et N^{K-1}, en utilisant pour chaque partie un identifiant $Id(P_{i_k})$ compris entre 0 et $N-1$.

On utilise la fonction modulo $Hash(n) = n \bmod S$ comme base pour la fonction de hachage, avec $S \geq N$ un nombre premier représentant la taille de la table de hachage. La fonction de hachage $Hash(Id(C_{i_1 i_2 \ldots i_K})) = Hash(\sum_{k=0}^{K-1} Hash(Id(P_{i_k}^{(k)})) Hash(N)^k)$ résultante est incrémentale vis à vis des changements de partie d'une signature.

Tout déplacement d'une cellule vers une partie adjacente implique un seul changement de partie dans sa signature, et la clé de hachage résultant se recalcule en O(1) au lieu de O(K) pour une fonction de hachage non incrémentale. Cela permet de gagner un facteur K dans les algorithmes d'optimisation des grilles, qui sont basés sur l'évaluation massive de fusions de parties adjacentes, impliquant des tests de "collision" entre cellules ne différent que par une seule partie.

3.2.3 Algorithme glouton

L'algorithme 3.2 généralise au cas multivarié l'heuristique gloutonne ascendante 3.1 décrite dans le cas univarié. On part de la grille initiale la plus fine, basée sur le produit cartésien des partition univariées comportant autant de parties élémentaires que d'individus. On évalue pour chaque variable toutes les fusions possibles entre parties adjacentes. La meilleure fusion est effectuée si elle améliore le coût de la partition, et l'algorithme est réitéré tant qu'il y a amélioration du coût.

Durant l'algorithme 3.2, la grille initiale comportant O(N^K) cellules est transformée en O(KN) étapes de fusions de parties en une grille finale comportant O(1) cellules. A chaque étape, O(KN) grilles sont évaluées, ce qui fait un total de O(K^2N^2) grilles évaluées durant le déroulement complet de l'algorithme.

En exploitant l'additivité du critère et la faible densité des individus dans la grille, il est démontré dans [Boullé, 2007c] (reproduit en annexe D) que l'algorithme 3.2 peut être implémenté en O($N \log N$) dans le cas bivarié. Dans le cas multivarié, en tenant compte de K dans le calcul de complexité, cette implémentation se généralise avec une complexité algorithmique de O($K^2 N \log N$).

De façon plus précise ces complexités algorithmiques correspondent aux cas de variables toutes numériques. Dans le cas de variables catégorielles, on se limite aux grilles impliquant au plus $\sqrt{N}$ groupes de valeurs. L'algorithme 3.2 a alors une complexité algorithmique de O($K^2 N \sqrt{N} \log N$) dans le cas de variables catégorielles.

3.2.4 Post-optimisation

Comme dans le cas univarié, des post-optimisations sont utilisées pour améliorer les grilles issue de l'heuristique gloutonne 3.2.

Post-optimisation de la taille de la grille. Afin de sortir d'un éventuel optimum local au moyen de plusieurs fusions successives, la procédure de fusion des parties de l'al-

Algorithme 3.2 Heuristique d'optimisation gloutonne ascendante d'une grille

Entrées: M {Grille initiale}
Sorties: $M^*, c(M^*) \leq c(M)$ {Grille finale optimisée}

$M^* \leftarrow M$
Tant que amélioration **faire**
  {Recherche de la meilleure amélioration}
  $M^+ \leftarrow M^*$
  Pour tout Variable X de la grille **faire**
    Pour tout Fusion m entre deux parties adjacentes de la partition de X **faire**
      {Evaluation de l'impact de la fusion m sur la partition M^*}
      $M' \leftarrow M^* + m$
      Si $c(M') < c(M^+)$ **alors**
        $M^+ \leftarrow M'$
      Fin si
    Fin pour
  Fin pour
  {Test d'amélioration}
  Si $c(M^+) < c(M^*)$ **alors**
    $M^* \leftarrow M^+$
  Fin si
Fin tant que

gorithme 3.2 est réitérée jusqu'à l'obtention d'une grille en une seule cellule. La meilleure grille rencontrée au cours de ces fusions "forcées" est retournée en sortie.

Post-optimisation de la grille. Pour post-optimiser une grille, on fige son partitionnement univarié sur $(K-1)$ dimensions et on optimise le partitionnement univarié sur la K^{ieme} dimension restante. En utilisant le fait qu'un critère additif de partitionnement multivarié est un critère additif de partitionnement univarié une fois $(K-1)$ dimensions fixées, les algorithmes de post-optimisation univariés introduits dans la section 3.1 sont réutilisables. Ainsi, les variables numériques sont post-optimisées en explorant des ajouts, suppressions ou redécoupages de frontières entre intervalles, et les variables catégorielles en déplaçant des valeurs entre groupes de valeurs.

3.2.5 Méta-heuristique

L'heuristique gloutonne 3.2 évalue $\mathrm{O}(K^2N^2)$ grilles parmi $\mathrm{O}(N^{2^K})$ avec un temps de calcul de $\mathrm{O}(K^2N\log N)$. Comme l'heuristique gloutonne est efficace en temps de calcul, il est naturel de l'appliquer de façon répétée afin de mieux explorer l'espace de recherche. La solution la plus simple est d'appliquer l'heuristique gloutonne plusieurs fois en partant de grilles initiales aléatoires. On applique ici la méta-heuristique de recherche à voisinage variable (VNS = Variable Neighborhood Search) [Hansen and Mladenovic, 2001], légèrement plus sophistiquée.

La mise en oeuvre de cette méta-heuristique est détaillée en annexe D dans le cas

bivarié. Pour une grille courante donnée, une nouvelle grille candidate est générée aléatoirement dans une voisinage de la grille courante. La grille candidate est optimisée (et post-optimisée), et devient la grille courante s'il y a amélioration. On recommence avec une nouvelle grille candidate dans un voisinage de même taille s'il y a eu amélioration, de taille élargie sinon. La procédure est réitérée jusqu'à atteindre la taille maximum de voisinage.

La méta-heuristique VNS est implémentée sous la forme d'un algorithme AnyTime, dont le seul paramètre est le temps de calcul alloué à l'optimisation. La complexité de base de l'algorithme est celle de l'heuristique gloutonne utilisée. Tout temps de calcul supplémentaire alloué en supplément de l'heuristique gloutonne permet une meilleure exploration de l'espace de recherche.

Cas bivarié. Dans le cas des grilles bivariées, la taille du voisinage est basée sur le nombre de parties ajoutées aux partitions univariées de la grille.

Cas multivarié, petit nombre de variables. Dans le cas multivarié, des modifications sont apportées à la notion de voisinage d'une grille, en incorporant l'aspect sélection de variable. Principalement, des ajouts ou suppressions de variables sont considérés, dont le nombre croît avec la taille du voisinage.

Cas multivarié, grand nombre de variables. Une grille comportant $K_s \geq K_{Max} = \lceil \log_2 N \rceil$ variables sélectionnées (partitionnées en au moins deux parties) contient plus de N cellules (car $2^K_{Max} \geq N$) et en moyenne moins de un individu par cellule. En se fondant sur la pénalisation des grilles complexes pour les critères d'évaluation présentés dans le chapitre 2, on choisit de limiter la recherche de solution aux grilles de dimension inférieure à K_{Max}. L'application de l'heuristique gloutonne 3.2 a dans ce cas un coût algorithmique de $O(K^2_{Max} N \log N)$. En appliquant l'heuristique gloutonne à K/K_{Max} sous-ensembles de K_{Max} variables, toutes les variables sont analysées une fois. De ce fait, on peut considérer que la complexité de base utile de la méta-heuristique VNS est $O(KN \log N \max(K, \log N))$ dans le cas de variables numériques, $O(KN\sqrt{N} \log N \max(K, \log N))$ dans le cas de variables catégorielles.

3.2.6 Synthèse

L'optimisation d'un modèle en grille peut se résumer à une extension au cas multivarié des algorithmes de partitionnement univarié relatifs à la discrétisation ou au groupement de valeurs.

L'algorithme d'optimisation principal est une heuristique gloutonne ascendante, qui partant d'une grille initiale élémentaire, recherche itérativement les meilleures fusions de parties pour chaque variable. Des post-optimisations sont effectuées afin d'améliorer la solution obtenue, en exploitant de façon locale le voisinage de cette solution. L'algorithme principal (accompagné des post-optimisations) est répété en partant de plusieurs solutions initiales, issues de l'exploration d'un voisinage global de la meilleure solution.

Ces algorithmes peuvent être implémentés efficacement en se basant sur les caractéristiques suivantes :

- additivité du critère d'évaluation des grilles, qui se décompose en une somme de termes indépendants portant sur la dimension de la grille, les variables, les parties de variables et les cellules,
- faible densité des individus dans la grille, qui comporte $O(N^K)$ cellules pour au plus N cellules non vides,
- utilisation de fonctions de hachage incrémentales, qui permettent de détecter les collisions de cellules entre parties adjacentes en $O(1)$.

L'exploitation systématique de ces caractéristiques conduit à des algorithmes relativement sophistiqués dans leur implémentation, mais permettant d'atteindre les performance suivantes :

- occupation mémoire en $O(KN)$ pour K variables et N individus,
- temps de calcul en $O(KN \log N \max(K, \log N))$ si toutes les variables sont numériques,
- temps de calcul en $O(KN\sqrt{N} \log N \max(K, \log N))$ dans le cas général de variables numériques ou catégorielles ayant de grands nombres de valeurs ($V \geq \sqrt{N}$).

3.3 Application à l'optimisation des modèles en grille de données

Cette section étudie l'applicabilité des algorithmes de partitionnement univarié et multivarié présentés en sections 3.1 et 3.2 pour l'optimisation des critères d'évaluation des modèles en grille présentés dans le chapitre 2.

3.3.1 Classification supervisée

La section 2.1 du chapitre 2 a introduit des critères d'évaluation pour les modèles en grille de classification supervisée, dans le cas univarié (discrétisation et groupement de valeurs) et dans le cas multivarié. On montre ici que tous ces critères sont additifs, ce qui permet d'appliquer les algorithmes d'optimisation introduits en section 3.1 dans le cas univarié et en section 3.2 dans le cas multivarié.

3.3.1.1 Discrétisation supervisée

Dans le cas de la classification supervisée, le nombre J de valeurs de la variable à expliquer, fixe et connu à l'avance, peut être traité comme une constante du problème. Pour chaque intervalle explicatif, le vecteur des effectifs par valeur à expliquer peut être considéré comme ne dépendant que de l'intervalle.

Dans ce contexte, le critère d'évaluation 2.4 d'une partition en intervalles

$$c(M) = \log N + \log \binom{N+I-1}{I-1} + \sum_{i=1}^{I} \log \binom{N_{i.}+J-1}{J-1} + \sum_{i=1}^{I} \log \frac{N_{i.}!}{N_{i1}!N_{i2}!\ldots N_{iJ}!}$$

peut s'écrire sous la forme additive

$$c(M) = c^{(V)}(X, I) + \sum_{i=1}^{I} c^{(P)}(P_i),$$

où

- $c^{(V)}(X, I) = \log N + \log \binom{N+I-1}{I-1}$ ne dépend que des caractéristiques de la variable X à discrétiser et du nombre d'intervalles I,
- $c^{(P)}(P_i) = \log \binom{N_{i.}+J-1}{J-1} + \log \frac{N_{i.}!}{N_{i1}!N_{i2}!\ldots N_{iJ}!}$ ne dépend que des caractéristiques de l'intervalle P_i, à savoir ses effectifs totaux $N_{i.}$ et par valeur à expliquer N_{iJ}.

Remarque 3.1. *L'additivité du critère de discrétisation supervisée provient de l'a priori proposé sur l'espace des modèles de discrétisation, principalement de l'exploitation de la hiérarchie des paramètres et de l'hypothèse d'indépendance des paramètres entre les intervalles. Cette remarque est valide pour toutes les critères d'évaluation des modèles en grille établis au chapitre 2.*

3.3.1.2 Groupement de valeurs supervisé

Dans le cas du groupement de valeurs supervisé, le nombre V de valeurs de la variable explicative est connu et peut être traité comme une constante du problème, comme N et J. Dans ce contexte, le critère d'évaluation 2.7 d'une partition en groupes de valeurs

$$c(M) = \log N + \log B(V, I) + \sum_{i=1}^{I} \log \binom{N_{i.} + J - 1}{J - 1} + \sum_{i=1}^{I} \log \frac{N_{i.}!}{N_{i1}!N_{i2}! \ldots N_{iJ}!}$$

peut s'écrire sous la forme additive où

- $c^{(V)}(X, I) = \log N + \log B(V, I)$ ne dépend que des caractéristiques de la variable X à grouper et du nombre de groupes I,
- $c^{(P)}(P_i) = \log \binom{N_{i.}+J-1}{J-1} + \log \frac{N_{i.}!}{N_{i1}!N_{i2}!\ldots N_{iJ}!}$.

3.3.1.3 Grille multivariée de classification supervisée

On vérifie que le critère d'évaluation 2.13 d'un modèle de classification supervisé M peut s'écrire sous la forme additive

$$\begin{aligned} c(M) = c^{(G)}(\mathcal{I}) &+ \sum_{k=1}^{K} c^{(V)}(X_k, I_k) \\ &+ \sum_{k=1}^{K} \sum_{i_k=1}^{I_k} c^{(P)}\left(P_{i_k}^{(k)}\right) + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_k=1}^{I_K} c^{(C)}(C_{i_1 i_2 \ldots i_K}) \end{aligned}$$

où

- $c^{(G)}(\mathcal{I}) = \log(K + 1) + \log \binom{K+K_s-1}{K_s}$ avec $K_s = \sum_{k=1}^{K} \mathbb{1}_{\{I_k > 1\}}$, ne dépend que des tailles $I_1, I_2, \ldots, I_K$ des partition univariées de la grille explicative,

- $c^{(V)}(X_k, I_k)$ ne dépend que des caractéristiques de la variable X_k et de la taille I_k de sa partition univariée :
  - . $\log N + \log \binom{N+I_k-1}{I_k-1}$ pour une variable numérique sélectionnée,
  - . $\log V_k + \log B(V_k, I_k)$ pour une variable catégorielle sélectionnée,
  - . 0 pour une variable non sélectionnée ($I_k = 1$),
- $c^{(P)}(P_{i_k}^{(k)}) = 0$ pour la partie $P_{i_k}^{(k)}$,
- $c^{(V)}(C_{i_1 i_2 \dots i_K}) = \log \binom{N_{i_1 i_2 \dots i_K}+J-1}{J-1} + (\log N_{i_1 i_2 \dots i_K}! - \sum_{j=1}^{J} \log N_{i_1 i_2 \dots i_K j}!)$ ne dépend que des caractéristiques de la cellule $C_{i_1 i_2 \dots i_K}$ et est nul pour les cellules vides.

3.3.2 Estimation de densité jointe

La section 2.4 du chapitre 2 a introduit un critère d'évaluation pour les modèle en grille d'estimation de densité jointe. On montre ici que ce critère est additif, ce qui permet d'appliquer l'algorithme d'optimisation introduit en section 3.2 dans le cas multivarié.

Pour chaque variable à expliquer catégorielle Y_k, le nombre V_k de valeurs est supposé connu. De ce fait, une fois une partition en groupes de valeurs spécifiée, le nombre $m_{j_k}^{(k)}$ de valeurs d'un groupe de valeurs $P_{j_k}^{(k)}$ peut être considéré comme ne dépendant que du groupe de valeurs. Les effectifs par valeur $n_{v_k}^{(k)}$ de chaque variable catégorielle sont invariants quelque soit le modèle en grille. Ils peuvent être traités comme une constante globale du problème, au même titre que le nombre d'individus N ou de variables K.

Dans ce contexte, on vérifie que le critère d'évaluation 2.37 d'un modèle d'estimation de densité jointe M peut s'écrire sous la forme additive

$$c(M) = c^{(G)}(\mathcal{I}) + \sum_{k=1}^{K} c^{(V)}(Y_k, J_k) + \sum_{k=1}^{K} \sum_{j_k=1}^{J_k} c^{(P)}\left(P_{j_k}^{(k)}\right) + \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \dots \sum_{j_K=1}^{J_K} c^{(C)}(C_{j_1 j_2 \dots j_K})$$

où

- $c^{(G)}(\mathcal{I}) = \log(K+1) + \log \binom{K+K_s-1}{K_s} + \log \binom{N+G_Y-1}{G_Y-1} + \log N! - \sum_{k \in \mathbb{K}_2} \sum_{v_k=1}^{V_k} \log n_{v_k}^{(k)}!$ avec $K_s = \sum_{k=1}^{K} \mathbb{1}_{\{J_k>1\}}$ et $G_Y = \prod_{k=1}^{K} J_k$, ne dépend que des tailles $J_1, J_2, \dots, J_K$ des partitions univariées de la grille à expliquer,
- $c^{(V)}(Y_k, J_k)$ ne dépend que des caractéristiques de la variable Y_k et de la taille J_k de sa partition univariée :
  - . $\log N$ pour une variable numérique sélectionnée,
  - . $\log V_k + \log B(V_k, J_k)$ pour une variable catégorielle sélectionnée,
  - . 0 pour une variable non sélectionnée ($J_k = 1$),
- $c^{(P)}(P_{j_k}^{(k)})$ ne dépend que des caractéristiques de la partie $P_{j_k}^{(k)}$:

. $\log N_{j_k}^{(k)}!$ pour une variable numérique,

. $\log N_{j_k}^{(k)}! + \log \binom{N_{j_k}^{(k)} + m_{j_k}^{(k)} - 1}{m_{j_k}^{(k)} - 1}$ pour une variable catégorielle,

– $c^{(V)}(C_{j_1 j_2 \dots j_K}) = -\log N_{j_1 j_2 \dots j_K}!$ ne dépend que des caractéristiques de la cellule $C_{j_1 j_2 \dots j_K}$ et est nul pour les cellules vides.

3.3.3 Cas général

L'optimisation des modèles en grille n'a pas été traité de façon systématique dans le cas de la régression, de la classification supervisée avec groupement des valeurs à expliquer, ou dans le cas général d'un nombre quelconque de variables explicatives ou à expliquer. On indique dans cette section les adaptations effectuées dans le cas de la régression univariée et on décrit une extension des algorithmes de partitionnement multivarié dans le cas des modèles en grille les plus généraux.

3.3.3.1 Régression univariée

Dans le cas d'une variable explicative numérique, un modèle de régression en grille se base sur une partition en I intervalles de la variable explicative X et une partition en J intervalles de la variable à expliquer Y. Le critère d'évaluation 2.15 correspondant

$$2\log N + \log \binom{N+I-1}{I-1} + \sum_{i=1}^{I} \log \binom{N_{i.}+J-1}{J-1} + \sum_{i=1}^{I} \log \frac{N_{i.}!}{N_{i1}!N_{i2}!\dots N_{iJ}!} + \sum_{j=1}^{J} \log N_{.j}!$$

n'est pas additif. En utilisant une grille bivariée, il peut s'écrire sous la forme "presque additive"

$$\begin{aligned} c(M) &= c^{(V)}(X, I) + c^{(V)}(Y, J) \\ &\quad + \sum_{i=1}^{I} c^{(P)}\left(P_i^{(X)}, J\right) + \sum_{j=1}^{J} c^{(P)}\left(P_j^{(Y)}\right) + \sum_{i=1}^{I}\sum_{j=1}^{J} c^{(C)}(C_{ij}) \end{aligned} \tag{3.3}$$

où

– $c^{(V)}(X, I) = \log N + \log \binom{N+I-1}{I-1}$ est le coût de la variable explicative X,

– $c^{(V)}(Y, J) = \log N$ est le coût de la variable à expliquer Y,

– $c^{(P)}(P_i^{(X)}, J) = \log \binom{N_{i.}+J-1}{J-1} + \log N_{i.}!$ est le coût de l'intervalle explicatif $P_i^{(X)}$,

– $c^{(P)}(P_j^{(Y)}) = \log N_{.j}!$ est le coût de l'intervalle à expliquer $P_j^{(Y)}$,

– $c^{(V)}(C_{ij}) = -\log N_{ij}!$ est le coût de la cellule C_{ij} de la grille.

Le problème provient du coût $c^{(P)}(P_i^{(X)}, J)$ de chaque intervalle explicatif, qui dépend de la taille de la partition de la variable à expliquer. Une façon simple de résoudre ce problème est de fixer temporairement la taille J quand on optimise la partition de la variable explicative. En choisissant de se limiter aux tailles $J \leq \sqrt{N}$ dans les algorithmes d'optimisation des grilles, on limite à $\sqrt{N}$ fois le nombre de réactualisations de la structure

de coût de la grille, chaque réactualisation impliquant $\mathrm{O}(N)$ coûts de partie à prendre en compte. L'impact sur le temps de calcul des algorithmes est alors limité à $\mathrm{O}(N\sqrt{N})$, ce qui ne dépasse pas le cas des grilles contenant des variables catégorielles.

La même solution s'applique au cas d'une variable explicative catégorielle pour la prédiction d'une variable à expliquer numérique.

Cette adaptation des méthodes d'optimisation n'a pas encore été réalisée complètement. Dans l'évaluation des méthodes de régression univariée conduite au chapitre 4, l'optimisation est effectuée uniquement en adaptant la post-optimisation, en partant de grilles de discrétisation bivariées aléatoires et en alternant l'optimisation de chaque partition univariée, l'autre étant fixée.

3.3.3.2 Estimation de densité conditionnelle

Dans le cas général de l'estimation de densité conditionnelle présenté en section 2.5, les modèles en grille se décomposent en une *grille explicative* de G_X cellules pour l'ensemble $\mathbb{K}_X$ des variables explicatives et une *grille à expliquer* de G_Y cellules pour l'ensemble $\mathbb{K}_Y$ des variables à expliquer.

Les individus de chaque cellule explicative sont distribués sur les G_Y cellules de la grille à expliquer. On peut également considérer que les individus de l'échantillon sont distribués sur la *grille complète* de $G = G_X G_Y$ cellules pour l'ensemble $\mathbb{K} = \mathbb{K}_X \cup \mathbb{K}_Y$ de toutes les variables.

Notons

- $X_1, X_2, \ldots, X_{K_X}, X_{K_X+1}, \ldots, X_K$ la liste de toutes les variables explicatives et à expliquer,
- $\mathcal{I}_X = (I_1, I_2, \ldots, I_{K_X})$ les dimensions de la grille explicative et
- $\mathcal{I} = (I_1, I_2, \ldots, I_{K_X}, I_{K_X+1}, \ldots, I_K)$ les dimensions de la grille complète.

En redistribuant les termes du critère d'évaluation 2.42 sur la grille explicative et la grille complète, on obtient une décomposition presque additive sous la forme

$$\begin{aligned} c(M) = & \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_k=1}^{I_{K_X}} c_X^{(C)} \left(C_{i_1 i_2 \ldots i_K}, G_Y \right) \\ & + c^{(G)}(\mathcal{I}) + \sum_{k \in \mathbb{K}} c^{(V)}(X_k, I_k) \\ & + \sum_{k \in \mathbb{K}} \sum_{i_k=1}^{I_k} c^{(P)} \left(P_{i_k}^{(k)} \right) + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_{k_X}=1}^{I_{K_X}} \cdots \sum_{i_k=1}^{I_K} c^{(C)} \left(C_{i_1 i_2 \ldots i_{K_X} \ldots i_K} \right) \end{aligned} \tag{3.4}$$

où pour la grille explicative :

- les coûts de grille $c_X^{(G)}$, de variable $c_X^{(V)}$ et de partie $c_X^{(P)}$ sont nuls,
- $c_X^{(V)}(C_{i_1 i_2 \ldots i_{K_X}}, G_Y) = \binom{N_{i_1 i_2 \ldots i_{K_X}} + G_Y - 1}{G_Y - 1} + \log N_{i_1 i_2 \ldots i_{K_X}}!$ dépend à la fois des caractéristiques de la cellule explicative $C_{i_1 i_2 \ldots i_{K_X}}$ et de la taille G_Y de la grille à expliquer, et est nul pour les cellules explicatives vides.

et où pour la grille complète :

- $c^{(G)}(\mathcal{I}) = \log(K+1) + \log\binom{K+K_s-1}{K_s} - \sum_{k\in\mathbb{K}_2\cap\mathbb{K}_Y}\sum_{v_k=1}^{V_k}\log n_{v_k}^{(k)}!$
  avec $K_s = \sum_{k=1}^{K}\mathbb{1}_{\{I_k>1\}}$, ne dépend que des tailles des partitions univariées,
- $c^{(V)}(X_k, I_k)$ ne dépend que des caractéristiques de la variable X_k et de la taille I_k de sa partition univariée :
  . $\log N + \binom{N+I_k-1}{I_k-1}$ pour une variable numérique explicative sélectionnée,
  . $\log N$ pour une variable numérique à expliquer sélectionnée,
  . $\log V_k + \log B(V_k, I_k)$ pour une variable catégorielle sélectionnée,
  . 0 pour une variable non sélectionnée,
- $c^{(P)}(P_{i_k}^{(k)})$ ne dépend que des caractéristiques de la partie $P_{i_k}^{(k)}$:
  . $\log N_{i_k}^{(k)}!$ pour une variable numérique à expliquer,
  . $\log N_{i_k}^{(k)}! + \log\binom{N_{i_k}^{(k)}+m_{i_k}^{(k)}-1}{m_{i_k}^{(k)}-1}$ pour une variable catégorielle à expliquer,
- $c^{(C)}(C_{i_1 i_2 \ldots i_{K_X} \ldots i_K}) = -\log N_{i_1 i_2 \ldots i_{K_X} \ldots i_K}!$ ne dépend que des caractéristiques de la cellule $C_{i_1 i_2 \ldots i_{K_X} \ldots i_K}$ et est nul pour les cellules vides.

Le critère $c(M)$ est donc la somme d'un critère presque additif pour la grille explicative et d'un critère additif pour la grille complète. Le couplage entre variables explicatives et à expliquer se traduit par la dépendance à la taille G_Y de la grille à expliquer dans les coûts de cellule explicative $c_X^{(V)}$.

Ce type de critère peut être optimisé en maintenant les deux structures de grilles explicative et complète en parallèle.

Pour toute fusion de partie d'une variable explicative, il faut maintenir simultanément les deux grilles en parallèle, ce qui ne fait que doubler le temps de calcul par rapport à une grille standard.

Pour toute fusion de partie d'une variable à expliquer, seule la grille complète est à maintenir. Par contre, il faut réactualiser les coûts sur l'ensemble de toutes les cellules explicatives, au nombre de N au maximum. Il faut également réactualiser les coûts de fusions de parties par variable explicative et retrier toutes ces fusions, ce qui implique au total $\mathrm{O}(K_X N \log N)$ opérations. En se limitant à des tailles de partition $I_k \leq \sqrt{N}$ pour chaque variable à expliquer, ces opérations de maintenance de la grille explicative sont à effectuer au plus $K_Y\sqrt{N}$ fois. L'impact total sur la complexité algorithmique de l'heuristique gloutonne 3.2 est donc $\mathrm{O}(K_X K_Y N\sqrt{N}\log N)$. En se limitant comme précédemment à des sélections de $K_s \leq \lceil \log_2 N \rceil$ variables dans la méta-heuristique VNS, on retrouve la même complexité algorithmique que dans le cas standard.

Il est donc possible d'optimiser efficacement le critère $c(M)$ dans le cas général de l'estimation de densité conditionnelle pour un nombre quelconque de variables explicatives et à expliquer. La complexité algorithmique est la même que dans le cas des critères additifs, à savoir $\mathrm{O}(KN\sqrt{N}\log N \max(K, \log N))$.

Cette adaptation des méthodes d'optimisation n'a pas encore été réalisée.

3.4 Conclusion

Nous avons introduit dans ce chapitre la notion d'additivité pour les critères d'évaluation d'une grille de données pouvant se décomposer en somme de termes sur la dimension de la grille, les variables, les parties et les cellules.

Nous avons ensuite proposé un algorithme efficace d'optimisation de ces critères additifs, basé essentiellement sur la mise en oeuvre d'une heuristique gloutonne de fusion itérative des parties de variables d'une grille. Les résultats d'optimisation issus de l'heuristique gloutonne sont améliorés en exploitant un voisinage local d'une grille solution au moyen d'algorithmes de post-optimisation, et en explorant un voisinage global à l'aide d'une méta-heuristique de recherche à voisinage variable. Globalement, ces heuristiques ont une complexité algorithmique de $O(KN)$ en occupation mémoire et $O(KN\sqrt{N}\log N \max(K, \log N))$ en temps de calcul.

Nous avons enfin montré que les critères d'évaluation des grilles du chapitre 2 sont additifs dans le cas de la classification supervisée et de l'estimation de densité jointe, ce qui permet d'utiliser directement les algorithmes présentés. Dans le cas général de l'estimation de densité conditionnelle pour un nombre quelconque de variables explicatives ou à expliquer, le critère d'évaluation n'est plus additif et des adaptations ont été présentées pour étendre les heuristiques d'optimisation tout en gardant la même complexité algorithmique.

Les algorithmes présentés ont été implémentés dans le cas de la classification supervisé, de l'estimation de densité jointe, et partiellement de la régression univariée. Les performances de ces algorithmes sont évaluées dans le chapitre 4.

4

Evaluation des modèles en grilles de données

Sommaire

Après avoir introduit les modèles en grille dans le chapitre 2 et leur optimisation dans le chapitre 3, nous présentons l'évaluation de ces modèles.

Pour des raisons de commodité, nous désignerons par le terme générique MODL[1] l'ensemble des méthodes issues des modèles en grille. On parlera ainsi de l'approche MODL, de la discrétisation ou du groupement de valeurs MODL, des prétraitement univariés et bivariés MODL, et des méthodes MODL de façon générale.

La section 4.1 introduit quelques notions générales utiles par la suite, puis étudie les propriétés intrinsèques des modèles en grille en exploitant les critères d'évaluation analytiques établis au chapitre 2.

La section 4.2 évalue les méthodes MODL de prétraitement univarié et bivarié pour la classification supervisée. Cette section est essentiellement un résumé synthétique des annexes A sur la discrétisation supervisée, B sur le groupement de valeurs supervisé et C sur le prétraitement bivarié pour la classification supervisée.

La section 4.3 évalue l'utilisation des prétraitements univariés et bivariés MODL en classification supervisée, au moyen du classifieur Bayesien naïf. Des évolutions de ce classifieur sont également présentées, notamment en ce qui concerne la sélection de variables et le moyennage de modèles.

La section 4.4 présente l'utilisation des modèles en grille multivariés pour la construction de nouvelles méthodes d'apprentissage, en classification supervisée et non supervisée.

La section 4.5 résume les résultats de participation à plusieurs challenges internationaux, qui ont permis de confronter les méthodes proposées aux méthodes alternatives de l'état de l'art sur plusieurs problèmes difficiles.

La section 4.6 décrit l'outil Khiops, qui incorpore les méthodes d'apprentissage supervisé basées sur les modèles en grille, et présente son utilisation sur des problèmes de France Télécom, notamment pour la recherche de représentation dans les grandes bases de données.

Enfin la section 4.7 conclut ce chapitre.

[1]MODL signifie Minimum Optimized Description Length, et se réfère à l'utilisation de techniques de MODeL selection, notamment l'approche MDL [Rissanen, 1978], à la recherche d'une distribution a priori Optimisée pour la préparation des données, et à la conception d'heuristiques d'Optimisation performantes.

4.1 Évaluation analytique

Après avoir introduit quelques concepts préalables, cette section présente un critère normalisé d'évaluation des modèles : le taux de compression. Des propriétés particulières des modèles supervisés de discrétisation et groupement de valeurs sont ensuite présentés. Enfin, une étude des seuils limites d'apprentissage est proposée dans le contexte des modèles de classification supervisée avec variables explicatives numériques.

4.1.1 Définitions préliminaires

Cette section définit une terminologie et une notation pour quelques modèles particuliers, puis présente quelques propriétés pour ces modèles particuliers.

Définition 4.1. Modèle nul.
Dans une famille de modèles en grille, le modèle nul, noté $M_\emptyset$, est le modèle contenant une seule cellule, ou un seul intervalle ou groupe de valeurs dans le cas univarié.

Définition 4.2. Modèle maximal.
Dans une famille de modèles en grille, le modèle maximal, noté M_{Max}, est le modèle contenant pour chaque variable numérique autant d'intervalles que d'individus et pour chaque variable catégorielle autant de groupes que de valeurs.

Définition 4.3. Modèle optimal.
Dans une famille de modèles en grille, le modèle optimal, noté $M_\star$, est le modèle le plus probable connaissant les données (MAP=maximum a posteriori), c'est à dire celui minimisant le critère d'évaluation introduit au chapitre 2.

Définition 4.4. Coût d'un modèle.
Le coût $c(M)$ d'un modèle M est égal à la valeur de son critère d'évaluation sur l'échantillon de données en apprentissage.

Définition 4.5. Pureté d'un sous-ensemble d'individus.
Un sous-ensemble d'individus est dit pur s'il est associé à une seule et même valeur à expliquer. On parlera ainsi de valeur pure ou de groupe pur dans le cas catégoriel, d'intervalle pur dans le cas numérique, de cellule pure dans une grille.

On présente maintenant une propriété des modèles nuls dans le cas le plus général de l'estimation de densité conditionnelle d'un ensemble $\mathbb{K}_Y$ de variables à expliquer connaissant un ensemble $\mathbb{K}_X$ de variables explicatives, défini en section 2.5 du chapitre 2.

Théorème 4.1. *Le coût du modèle nul sur un échantillon de taille N est asymptotiquement égal à N fois l'entropie de Shannon des variables à expliquer quand $N \to \infty$.*

Démonstration. Dans le cas le plus général de l'estimation de densité conditionnelle, la grille ne contenant qu'une seule cellule du modèle nul est évaluée selon la formule 2.42 du

chapitre 2 :

$$c(M_\emptyset) = \log(K+1) + \sum_{k \in \mathbb{K}_2 \cap \mathbb{K}_Y} \log \binom{N + V_k - 1}{V_k - 1} + \sum_{k \in \mathbb{K}_1 \cap \mathbb{K}_Y} \log N! + \sum_{k \in \mathbb{K}_2 \cap \mathbb{K}_Y} \log \frac{N!}{n_{v_k}^{(k)}! n_{v_k}^{(k)}! \dots n_{V_k}^{(k)}!}.$$

On limite la démonstration au cas des variables catégorielles, pour lesquelles on possède une définition de l'entropie de Shannon $H(Y_k)$. Pour V_k valeurs $y_1, y_2, \dots, y_{V_k}$, l'entropie est égale à

$$H(Y_k) = -\sum_{v_k=1}^{V_k} P(y_{v_k}) \log P(y_{v_k}).$$

En assimilant cette entropie à son estimation par les probabilités empiriques, on a :

$$H(Y_k) = -\sum_{v_k=1}^{V_k} \left(\frac{n_{v_k}^{(k)}}{N} \right) \log \left(\frac{n_{v_k}^{(k)}}{N} \right).$$

On revient maintenant au coût $c(M_\emptyset)$ du modèle nul en utilisant l'approximation $\log N = N(\log N - 1) + O(\log N)$ quand $N \to \infty$, basée sur la formule de Stirling.

Pour les termes binomiaux, on a

$$\begin{aligned} \log \binom{N + V_k - 1}{V_k - 1} &= (N + V_k - 1)(\log(N + V_k - 1) - 1) - N(\log N - 1) + O(\log N), \\ &= (V_k - 1) \log N + O(\log N). \end{aligned}$$

Pour les termes multinômiaux correspondant aux variables catégorielles, on a

$$\begin{aligned} \log \frac{N!}{n_{v_k}^{(k)}! n_{v_k}^{(k)}! \dots n_{V_k}^{(k)}!} &= N(\log N - 1) - \sum_{v_k=1}^{V_k} n_{v_k}^{(k)} \left(\log n_{v_k}^{(k)} - 1 \right) + O(\log N), \\ &= N \log N - \sum_{v_k=1}^{V_k} n_{v_k}^{(k)} \log n_{v_k}^{(k)} + O(\log N), \\ &= -\sum_{v_k=1}^{V_k} n_{v_k}^{(k)} \left(\log n_{v_k}^{(k)} - \log N \right) + O(\log N), \\ &= N H(Y_k) + O(\log N). \end{aligned}$$

En utilisant ces approximations, on obtient

$$\begin{aligned} c(M_\emptyset) &= \log(K+1) + \sum_{k \in \mathbb{K}_2 \cap \mathbb{K}_Y} (V_k - 1) \log N \\ &\quad + \sum_{k \in \mathbb{K}_1 \cap \mathbb{K}_Y} \log N! + \sum_{k \in \mathbb{K}_2 \cap \mathbb{K}_Y} N H(Y_k) + O(\log N), \\ &= \sum_{k \in \mathbb{K}_1 \cap \mathbb{K}_Y} \log N! + \sum_{k \in \mathbb{K}_2 \cap \mathbb{K}_Y} N H(Y_k) + O(\log N). \end{aligned}$$

□

Remarque 4.1. *Dans le cas d'une variable numérique Y traitée dans le cadre de la statistique d'ordre, l'entropie n'est pas définie dans la littérature à notre connaissance. Si on réduit une variable de rang à une variable catégorielle ayant N valeurs toutes de fréquence $1/N$, on obtient pour l'entropie :*

$$H(Y) = -\sum_{n=1}^{N} \left(\frac{1}{N}\right) \log \left(\frac{1}{N}\right) = \log N.$$

Une alternative est de considérer que l'entropie de l'ensemble des individus pour une variable de rang est une mesure de l'incertitude sur l'ordonnancement des N individus. Comme il y a $N!$ ordonnancements possibles, on obtient dans ce cas une entropie globale de $\log N! = N(\log N - 1) + O(\log N)$, soit une entropie par individu de $(\log N - 1)$. En adoptant cette deuxième approche pour la définition d'une entropie sur une variable de rang, on étend la démonstration du théorème 4.1 au cas des variables numériques.

4.1.2 Taux de compression d'un modèle

On propose dans cette section d'utiliser les coûts de modèles comme critères d'évaluation des variables ou des sous-ensembles de variables, et on en présente une normalisation appelée taux de compression.

Le coût $c(M)$ d'un modèle en grille correspond à la probabilité a posteriori que le modèle M décrive les données à expliquer observées. Plus précisément, ce coût résulte du calcul analytique $c(M) = -\log P(M) - \log P(D|M)$. Puisque les log négatifs de probabilités correspondent à des longueurs de codage [Shannon, 1948], le critère $c(M)$ s'interprète comme la capacité d'un modèle en grille à encoder les valeurs à expliquer connaissant les valeurs explicatives. Le coût $c(M_\emptyset)$ du modèle nul représente la longueur de codage des valeurs à expliquer quand aucune information explicative n'est utilisée. Selon le théorème 4.1, $c(M_\emptyset)$ est asymptotiquement égal à N fois l'entropie de Shannon des variables à expliquer. Les modèles plus complexes que le modèle nul comportent plusieurs cellules, ce qui offre la possibilité de coder plus efficacement les données à expliquer, en exploitant l'entropie localement aux cellules. Des cellules de petite taille permettent d'identifier des régions où l'entropie est faible (distribution déséquilibrée des valeurs à expliquer), mais les modèles trop complexes sont pénalisés par une longueur de codage accrue pour la description des paramètres du modèle.

En se basant sur ces interprétations probabiliste et de longueur de codage, on propose d'utiliser les coûts des modèles pour les comparer entre eux. Dans le cas univarié, cela fournit un critère d'évaluation des variables, permettant de les ordonner par importance prédictive décroissante. Dans le cas multivarié, cela permet de comparer de façon équitable plusieurs sous-ensembles de variables.

De façon à obtenir un indicateur normalisé, on définit le taux de compression $g(M)$ d'un modèle par la transformation suivante de son coût $c(M)$:

$$g(M) = 1 - \frac{c(M)}{c(M_\emptyset)}. \tag{4.1}$$

Le taux de compression $g(M)$ prend des valeurs comprises entre 0 et 1 pour tout modèle meilleur que le modèle nul (sinon, $g(M)$ est négatif). Le taux de compression vaut

0 pour le modèle nul et il est maximal pour le modèle optimal, c'est à dire pour le modèle de description des données à expliquer le plus probable conditionnellement aux données explicatives. Il ne peut atteindre asymptotiquement la valeur 1 que dans le cas de valeurs à expliquer parfaitement séparables par un modèle en grille.

4.1.3 Propriétés des critères dans le cas univarié

Cette section présente quelques propriétés dans le cas des méthodes MODL univariées pour la classification supervisée. Ces propriétés sont démontrées en annexe A pour la discrétisation supervisée et en annexe B pour le groupement de valeurs supervisé.

4.1.3.1 Discrétisation supervisée univariée

Le théorème 4.2 démontre que les frontières des intervalles optimaux ne peuvent séparer des individus consécutifs ayant même valeur à expliquer. Cette propriété a déjà été démontrée dans le cas de la méthode de discrétisation MDLPC [Fayyad and Irani, 1992] et pour une plus large gamme de mesures d'impureté utilisées comme critère de discrétisation [Elomaa and Rousu, 1996].

Cette propriété est intéressante pour améliorer l'efficacité des algorithme d'optimisation. En effet, seules les frontières entre sous-séquences pures d'individus sont à considérer pour trouver une discrétisation de qualité, ce qui évite de traiter les $N-1$ frontières potentielles entre les individus de l'échantillon.

Théorème 4.2. *Dans un modèle optimal de discrétisation supervisée univariée, deux individus consécutifs (selon la variable numérique explicative) ayant la même valeur a expliquer sont nécessairement dans le même intervalle.*

Les théorèmes 4.3 et 4.4 fournissent une validation intuitive de la pertinence des critères, en confirmant que des intervalles basés sur des singletons ne peuvent donner lieu à une bonne généralisation.

Théorème 4.3. *Dans un modèle optimal de discrétisation supervisée univariée, il n'existe pas deux intervalles consécutifs ne contenant qu'un seul individu.*

Théorème 4.4. *Dans le cas de la discrétisation supervisée univariée et pour une variable à expliquer booléenne, le modèle nul, ne contenant qu'un seul intervalle, est plus probable que le modèle maximal, contenant autant d'intervalles que d'individus.*

Remarque 4.2. *Le critère d'évaluation d'une discrétisation supervisée MODL n'exclut pas la possibilité d'intervalles vides, possibilité qui est dénombrée dans le terme d'a priori* $\log\binom{N+I-1}{I-1}$ *utilisé pour le choix des bornes des intervalles. Cette option est délibérée, afin de favoriser les discrétisations comportant peu d'intervalles. Si l'on exclut la possibilité d'intervalles vides, le terme d'a priori devient* $\log\binom{N-1}{I-1}$ *et les théorèmes 4.2, 4.3 et 4.4 ne sont plus vrais. Il est à noter qu'il n'est pas utile dans les algorithmes d'explorer les discrétisations comportant des intervalles vides, puisque l'ajout d'un intervalle vide entraîne toujours une augmentation du coût d'une discrétisation.*

4.1.3.2 Groupement de valeurs supervisé univarié

Dans le cas du groupement de valeurs supervisé univarié, les théorèmes 4.5 et 4.6 fournissent une première validation intuitive de la pertinence des critères.

Théorème 4.5. *Dans un modèle optimal de groupement de valeurs supervisé univarié, deux valeurs explicatives pures associées à la même valeur à expliquer sont nécessairement dans le même groupe de valeurs.*

Théorème 4.6. *Dans le cas du groupement de valeurs supervisé univarié et pour une variable à expliquer booléenne, le modèle nul, ne contenant qu'un seul groupe, est le modèle optimal pour toute variable explicative ayant autant de valeurs que d'individus.*

4.1.4 Seuils d'apprentissage

Cette section étudie les seuils d'apprentissage dans le cas des très faibles nombres d'individus, en s'intéressant à la classification supervisée pour des variables explicatives numériques.

4.1.4.1 Introduction

Si la fouille de données se focalise sur des problèmes pour lesquels les volumes de données sont très importants, il existe de nombreuses situations où le nombre d'individus disponibles est très faible. C'est notamment le cas en médecine pour le nombre de patients impliqués dans l'évaluation d'une nouvelle thérapie, ou en biostatistique dans le domaine des micro-arrays par exemple, caractérisés par quelques dizaines d'individus seulement pour des milliers de variables. Ces problèmes sont critiques vis-à-vis de la fiabilité des connaissances induites par les algorithmes d'apprentissage.

L'étude des seuils d'apprentissage permet de caractériser les comportements aux limites pour des échantillons de très petite taille. On se pose notamment la question du nombre minimal d'individus nécessaire pour détecter une information, c'est à dire pour que le modèle optimal ne soit pas réduit au modèle nul. On se pose également la question de savoir, pour un nombre d'individus et de variables donnés, quelle est la quantité d'information (mesurée en nombre de cellules) maximale détectable par un modèle en grille. On apporte ici quelques éléments de réponse, dans le cas de la classification supervisée pour des variables explicatives numériques.

Détection de grille informative. Les termes "détection d'information" sont trop génériques. On se place ici dans le contexte précis de la recherche d'information au moyen de modèles en grille, et on utilisera les termes *détection de grille informative* pour toute grille contenant au moins deux cellules non vides de coût strictement meilleur que celui du modèle nul. On parlera de grille informative de dimension K *minimale* pour une grille informative comportant au moins deux parties non vides par dimension, et de grille *maximale* pour une grille informative comportant un maximum de cellules (d'intervalles si $K = 1$).

4.1.4.2 Seuil de détection d'une discrétisation univariée informative minimale

Soit une variable catégorielle à expliquer booléenne de valeurs 0 et 1, et soit un échantillon de N individus (avec N_1 et N_2 individus par valeur à expliquer) et K variables explicatives numériques.

On étudie le cas où pour au moins une variable explicative numérique, les individus ordonnés suivant cette variable donnent lieu à une séquence séparable en deux intervalles purs. Pour cela, on calcule le coût du modèle nul $M_\emptyset$, réduit à un seul intervalle, et du modèle M_2, basé sur les deux intervalles séparant parfaitement les individus. Le modèle M_2 est *minimal* dans le sens où il possède un nombre d'intervalles minimum parmi les modèles non nuls.

Sans sélection de variables. En utilisant la formule 2.4 pour évaluer le coût de chaque modèle (en ignorant l'a priori de sélection de variables), on obtient

$$c(M_\emptyset) = \log N + \log(N+1) + \log \frac{N!}{N_1!N_2!}, \tag{4.2}$$

$$c(M_2) = \log N + \log(N+1) + \log(N_1+1) + \log(N_2+1). \tag{4.3}$$

En prenant un échantillon de $N = 6$ individus avec $N_1 = N_2 = 3$, on a $c(M_\emptyset) = \log 840$ et $c(M_2) = \log 672$, ce qui montre qu'un modèle $\boxed{000 \mid 111}$ à deux intervalles non vides est strictement meilleur que le modèle nul. Une recherche exhaustive sur tous les échantillons confirme que le nombre minimal d'individus nécessaire est bien six, soit seulement trois par intervalle.

Ce nombre d'individus est très faible. Il n'y a que $64 = 2^6$ séquences de 0 ou 1 possibles pour six individus, et donc une chance sur 64 d'obtenir la séquence 000111 par hasard. Quand le nombre de variables explicatives dépasse quelques dizaines, obtenir une grille informative par hasard devient probable, ce qui impose de tenir compte du nombre de variables.

Avec sélection de variables. En intégrant l'a priori de sélection de variables introduit dans la formule 2.13 dans le cas multivarié de la classification supervisée, le coût de chaque modèle est cette fois

$$c(M_\emptyset) = \log(K+1) + \log(N+1) + \log \frac{N!}{N_1!N_2!}, \tag{4.4}$$

$$c(M_2) = \log(K+1) + \log K + \log N + \log(N+1) + \log(N_1+1) + \log(N_2+1). \tag{4.5}$$

Une recherche exhaustive montre que 12 individus au minimum sont nécessaires pour détecter une discrétisation informative dans le cas d'une seule variable explicative. Il en faut 37 pour un million de variables explicatives, comme le montre la figure 4.1.

Afin de mieux caractériser le comportement visualisé sur la figure 4.1, on montre dans le théorème 4.7 comment quantifier l'impact de chaque ajout d'individu dans l'échantillon sur le seuil d'apprentissage d'une discrétisation minimale.

Théorème 4.7. *Dans le cadre de la discrétisation supervisée univariée pour la prédiction d'une variable à expliquer booléenne équidistribuée, l'ajout d'un seul individu permet de*

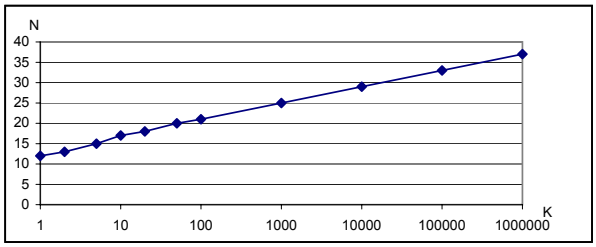


FIG. 4.1 – Nombre d'individus N minimum pour obtenir une discrétisation informative minimale, en fonction du nombre K de variables explicatives numériques.

doubler le nombre de variables de l'espace de recherche, tout en gardant la possibilité de détecter une discrétisation informative.

Démonstration. Notons $\delta(K, N, N_1)$ la différence de coût $c(M_2) - c(M_\emptyset)$ entre le modèle en deux intervalles et le modèle nul en un seul intervalle. Cette différence de coût permet de trouver le seuil de détection d'une discrétisation informative en fonction du nombre d'individu N et du nombre de variables K. On a

$$\delta(K, N, N_1) = \log K + \log N + \log(N_1 + 1) + \log(N_2 + 1) - \log \frac{N!}{N_1! N_2!}.$$

Si on ajoute un individu (pour la valeur 0 sans perte de généralité) et qu'on se donne cette fois K' variables, on obtient

$$\delta(K', N + 1, N_1 + 1) = \log K' + \log N + \log(N_1 + 2) + \log(N_2 + 1) - \log \frac{(N + 1)!}{(N_1 + 1)! N_2!},$$

soit encore

$$\delta(K', N + 1, N_1 + 1) = \delta(K, N, N_1) + \log K' - \log K + \log(N_1 + 2) - \log(N + 1).$$

Supposons que $\delta(K, N, N_1) < 0$, c'est à dire que le modèle M_2 représente une discrétisation informative pour N individus et K variables. Il suffit alors que

$$\log K' - \log K + \log(N_1 + 2) - \log(N + 1) \leq 0$$

pour que $\delta(K', N + 1, N_1 + 1) < 0$, donc pour qu'il existe une discrétisation informative pour $N + 1$ individus et K' variables. En ajoutant un individu, on peut donc continuer à détecter une discrétisation informative en augmentant le nombre de variables jusqu'à

$$K' \leq K \frac{N + 1}{N_1 + 2}, \tag{4.6}$$

ce qui correspond approximativement à un doublement du nombre de variables dans le cas de deux valeurs à expliquer équidistribuées. □

Remarque 4.3. *L'inégalité 4.6 dans la preuve du théorème 4.7 montre que l'ajout d'un individu apporte plus d'information s'il concerne la valeur minoritaire de la variable à expliquer.*

4.1.4.3 Seuil de détection d'une discrétisation univariée informative maximale

On s'intéresse maintenant au nombre d'individus minimum nécessaire pour détecter une discrétisation informative maximale, comportant le plus grand nombre d'intervalles possible pour un nombre d'individus donné.

Afin de trouver un majorant de ce seuil, on considère une variable à expliquer booléenne équidistribuée et une variable explicative numérique permettant d'ordonner les individus en une suite de I sous-séquences pures, associées alternativement aux valeurs 0 et 1, toutes de même effectif $N_I = N/I$.

On évalue le modèle nul $M_\emptyset$ et le modèle M_I de discrétisation en I intervalles d'effectifs N_I :

$$c(M_\emptyset) = \log(K+1) + \log(N+1) + \log \frac{N!}{N_1!N_2!}, \tag{4.7}$$

$$c(M_I) = \log(K+1) + \log K + \log N + \log \binom{N+I-1}{I-1} + I \log(N_I+1). \tag{4.8}$$

Pour chaque nombre d'intervalles I, on recherche par simulation numérique l'effectif minimum par intervalle aboutissant à un modèle M_I meilleur que le modèle nul. Le résultat confirme qu'il faut au moins 12 individus, soit six par intervalle pour $I = 2$, sept par intervalle pour $I = 3$, et huit par intervalle à partir de $I = 8$, et ce quelque soit le nombre d'intervalles (simulé jusqu'à 100 millions d'intervalles, nécessitant 800 millions d'individus). Cet effectif minimum constant égal à huit peut être confirmé de façon analytique, ce qui constitue la preuve (non fournie ici) du théorème 4.8 et de son corollaire 4.9 (en prenant $K = 1$, l'aspect sélection de variables étant marginal comme pour le théorème 4.7).

Théorème 4.8. *Dans le cadre de la discrétisation supervisée univariée pour la prédiction d'une variable à expliquer booléenne équidistribuée, le nombre d'individus minimum nécessaire pour la détection de I intervalles informatifs est au plus égal à 8 fois le nombre d'intervalles.*

Théorème 4.9. *Dans le cadre de la discrétisation supervisée univariée pour la prédiction d'une variable à expliquer booléenne équidistribuée, il suffit de 8 individus supplémentaires pour permettre d'accroître de 1 intervalle la complexité maximale d'une discrétisation informative.*

Comme seul un cas particulier de discrétisation a été considéré (et non toutes les discrétisations possibles), le résultat du théorème 4.8 fournit un majorant pour le nombre d'individus minimum nécessaire pour la détection d'une discrétisation informative maximale à I intervalles, dans le meilleur des cas. Autrement dit, si l'on dispose de N individus, il est fortement improbable de trouver une discrétisation informative comportant plus de $N/8$ intervalles.

4.1.4.4 Seuil de détection d'une grille de discrétisation multivariée informative minimale

On s'intéresse cette fois aux discrétisations multivariées informatives minimales, c'est à dire aux grilles de données pour lesquelles chaque variable explicative est discrétisée en exactement deux intervalles non vides. Ainsi, chaque dimension de la grille est utile, car non réduite à un seul intervalle.

Afin de trouver un majorant de ce seuil, on considère un jeu de données comportant K variables explicatives numériques et une variable à expliquer booléenne équidistribuée, parfaitement séparable au moyen d'un XOR de dimension $K_s \leq K$.

XOR en dimension 2

1	0
0	1

La discrétisation multivariée informative recherchée correspond à une grille basée sur une sélection de K_s variables explicatives, toutes discrétisées en deux intervalles (d'index 0 et 1) de même effectif. La grille correspondante contient $G = 2^{K_s}$ cellules, que l'on suppose toutes de même effectif $N_G = N/G$. Les cellules de cette grille sont associées à la valeur à expliquer 0 ou 1 selon la parité de la somme des index des intervalles par variable explicative.

On évalue le modèle nul $M_\emptyset$ et le modèle M_G de grille multivariée comportant K_s variables et G cellules d'effectif N_G.

$$c(M_\emptyset) = \log(K+1) + \log(N+1) + \log \frac{N!}{N_1!N_2!}, \tag{4.9}$$

$$c(M_G) = \log(K+1) + \log \binom{K+K_s-1}{K_s-1} + K_s \log N + K_s \log(N+1) + G \log(N_G+1). \tag{4.10}$$

Pour chaque nombre de dimension K_s de la grille, on recherche par simulation numérique l'effectif minimum par cellule aboutissant à un modèle M_G meilleur que le modèle nul. Le résultat indique qu'il faut au moins dix individus par cellule, soit quarante individus en tout en dimension 2. Cet effectif minimum par cellule décroît jusqu'à atteindre deux individus au minimum par cellule en dimension dix et au delà (testé jusqu'en dimension 1000, et visualisé jusqu'en dimension 20 sur la figure 4.2).

Cela signifie qu'il existe au moins une grille informative de dimension K_s qui est détectable avec 2^{K_s+1} individus. Cela conduit au théorème 4.10, dont la preuve n'est pas présentée ici.

Comme pour le théorème 4.7, le nombre total de dimensions K joue un rôle négligeable vis à vis du nombre de dimensions sélectionnées K_s pour la grille. Alors qu'il faut approximativement doubler le nombre d'individus pour ajouter une dimension à la grille informative, il suffit d'environ $K_s \log 2$ individus supplémentaires pour doubler la taille K de l'espace de représentation.

Théorème 4.10. *Dans le cadre de la discrétisation supervisée multivariée pour la prédiction d'une variable à expliquer booléenne équidistribuée, le nombre d'individus minimum nécessaire pour la détection d'une grille informative minimale de dimension K_s est asymptotiquement au plus égal à 2 fois le nombre 2^{K_s} de cellules.*

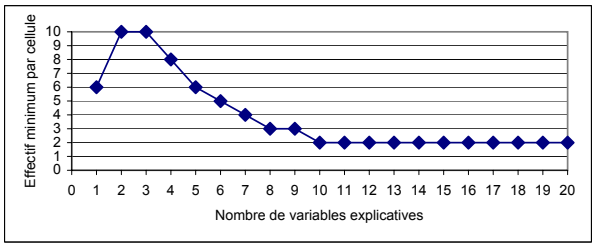


FIG. 4.2 – Nombre minimum N_G d'individus par cellule pour détecter un motif de type XOR au moyen d'une grille informative, en fonction du nombre de variables impliquées dans le XOR. Par exemple, pour un XOR en dimension 5, il faut 6 individus par cellule, soit $192 = 2^5 * 6$ individus dans l'échantillon pour que le XOR soit correctement identifié au moyen d'une grille de $32 = 2^5$ cellules.

Remarque 4.4. *De façon duale, ce théorème indique que pour un échantillon de N individus, il est très improbable d'identifier une grille informative de dimension supérieure à $\log_2 N$. Cela justifie le choix de se limiter aux grilles de dimension inférieure à $\log_2 N$ pour l'optimisation des grilles dans le cas multivarié (cf. section 3.2 pour l'optimisation des grilles dans le cas des grands nombres de variables).*

Remarque 4.5. *Selon la figure 4.2, le nombre d'individus nécessaire pour identifier un XOR complexe est faible, par exemple inférieur à 200 individus pour un XOR en dimension 5. Le nombre total K de variables a un rôle négligeable dans le coût $c(M_G)$ du modèle en grille, si bien qu'environ 220 individus seulement sont suffisants pour détecter le XOR de dimension 5 dans un espace de 1000 variables. Le critère reste donc capable de reconnaître la grille informative avec peu d'individus. En revanche, les méthodes d'optimisation atteignent leur limite. Étant donné que les algorithmes n'évaluent que des grilles de dimension inférieure à $10 \sim \log_2 1000$, et que la probabilité que 10 dimensions choisies au hasard parmi 1000 contiennent les 5 dimensions informatives est très faible, il est très peu probable de détecter le XOR par une grille informative au moyen des algorithmes présentés au chapitre 3.*

4.1.5 Bilan de l'évaluation analytique

Cette section démontre l'intérêt des critères analytiques d'évaluation des modèles en grille.

Ces critères permettent de mettre en évidence certaines propriétés des modèles, de façon exacte et non empirique. Dans le cas des discrétisations et groupements de valeurs univariés pour la classification supervisée, on a ainsi pu démontrer des propriétés permettant d'une part une validation "intuitive" des critères, d'autre part une amélioration des algorithmes d'optimisation, qui peuvent exploiter ces propriétés pour élaguer sans risque certaines parties de l'espace des modèles.

Les critères analytiques permettent également d'initier une étude des comportements aux limites, dans le cas des échantillons de très petite taille, ce qui est intéressant tant d'un point de vue théorique que d'un point de vue pratique pour déterminer les conditions d'analyse des jeux de données de très petite taille comportant de nombreuses variables

explicatives.

Bien que restreinte au cas des variables explicatives numériques pour la classification supervisée, cette étude des seuils d'apprentissage a permis de déterminer des majorants pour le nombre minimum d'individus nécessaire pour apprendre des concepts au moyen de modèles en grille. On synthétise ci-dessous les résultats obtenus en fournissant des ordres de grandeur pour les seuils d'apprentissage :

- une dizaine d'individus est nécessaire pour identifier une discrétisation informative minimale,
- une dizaine d'individus par intervalle est nécessaire pour identifier une discrétisation informative maximale,
- environ 2^K individus sont nécessaires pour identifier une grille informative minimale de dimension K,
- un individu supplémentaire permet de doubler le nombre de variables explicatives de l'espace de représentation pour rechercher une discrétisation informative,
- environ K individus supplémentaires permettent de doubler le nombre de variables explicatives de l'espace de représentation pour rechercher une grille informative de dimension K.

4.2 Prétraitements univariés et bivariés pour la classification supervisée

Les méthodes MODL de prétraitement univarié et bivarié permettent d'estimer l'importance prédictive de chaque variable ou paire de variables. Ces méthodes de prétraitement sont ici évaluées pour leur performance intrinsèque, indépendamment de leur utilisation dans un classifieur.

L'objectif de cette section est de résumer l'ensemble des expérimentations menées dans les articles [Boullé, 2006b, Boullé, 2005a, Boullé, 2007b], reproduits en annexes A, B et C. On présente également quelques expérimentations sur des jeux de données artificiels dans le cas de variables explicatives numériques, afin de compléter les résultats analytiques sur les seuils d'apprentissage.

4.2.1 Evaluation comparative de la discrétisation

Les sections 4 et 5 de l'article [Boullé, 2006b] (reproduit en annexe A) présentent une évaluation comparative intensive de la méthode MODL de discrétisation supervisée.

L'expérimentation compare les performances de la méthode MODL avec celles de 8 méthodes alternatives, sur la base de 15 jeux de données réels de l'UCI [Blake and Merz, 1996], représentant au total 181 variables explicatives numériques considérées individuellement. Les méthodes sont évaluées en mesurant le taux de bonne prédiction, la robustesse et le nombre d'intervalles des discrétisations.

Les résultats d'expérimentation montrent que la méthode MODL domine l'ensemble des méthodes alternatives sur chacun des critères évalués. Sur aucun des critères, la domination n'est stricte : d'autres méthodes sont aussi performantes en taux de prédiction, ou

en robustesse, mais toutes les méthodes alternatives (sauf une) sont dominées strictement sur au moins un critère d'évaluation.

Des expérimentations plus poussées sur la base de jeux de données artificiels permettent d'identifier les différences entre les méthodes. La méthode MODL est significativement plus performante que les méthodes alternatives quand on considère conjointement la résistance au bruit et la capacité de détection des motifs fins. Son domaine d'application n'est pas limité, et elle garde son efficacité tant dans le domaine des échantillons de petite taille que de grande taille.

4.2.2 Evaluation comparative du groupement de valeurs

La section 3 de l'article [Boullé, 2005a] (reproduit en annexe B) présente une évaluation comparative intensive de la méthode MODL de groupement de valeurs.

L'expérimentation compare les performances de la méthode MODL avec celles de 5 méthodes alternatives, sur la base de 12 jeux de données réels de l'UCI [Blake and Merz, 1996], représentant au total 230 variables explicatives catégorielles considérées individuellement, ainsi que 2614 nouvelles variables catégorielles construites par produit cartésien des variables initiales. Les méthodes sont évaluées en mesurant la qualité de l'estimation univariée de densité conditionnelle et le nombre de groupes des groupements de valeurs.

La méthode MODL est celle qui produit le moins de groupes tout en ayant la meilleure estimation de densité conditionnelle. En utilisant un groupement de valeurs en tant que prétraitement pour le classifieur Bayesien naïf, les résultats obtenus sont significativement meilleurs avec la méthode MODL qu'avec les méthodes alternatives.

Des expérimentations plus poussées sur la base de jeux de données artificiels montrent comme dans le cas numérique que la méthode MODL s'avère significativement plus performante que les méthodes alternatives quand on considère conjointement la résistance au bruit et la capacité de détection des motifs fins.

4.2.3 Evaluation des grilles bivariées

Les sections 5 et 6 de l'article [Boullé, 2007c] (reproduit en annexe C) présentent une évaluation expérimentale intensive de la méthode MODL d'évaluation supervisée des paires de variables.

L'expérimentation évalue les apports de la méthode MODL sur la base de 30 jeux de données réels de l'UCI [Blake and Merz, 1996]. L'évaluation met en évidence les apports en préparation des données, notamment l'utilisation du taux de compression pour comparer l'importance prédictive de chaque variable ou paire de variables, pour la détection des variables inutiles, redondantes ou en corrélation constructive. L'intelligibilité des modèles en grille est également illustrée, en ce qui concerne la visualisation des informations identifiées pour des paires de variables numériques, catégorielles ou mixtes, et l'interprétation de ces informations sous la forme de base de règles de décision.

En utilisant les prétraitements bivariés en tant que prétraitement pour le classifieur Bayesien naïf, les résultats démontrent une amélioration très significative des performances

quand l'ajout des nouvelles variables issues de l'analyse bivariée est couplée avec une méthode performante de sélection de variables.

Des expérimentations intensives sur la base de jeux de données artificiels confirment les résultats des expérimentations en univarié, avec une forte résistance au bruit dans le cas des paires de variables numériques ou catégorielles et une capacité de détection sensible des motifs complexes. Ces expérimentations complètent les résultats sur les seuils d'apprentissage, en montrant par exemple que 100000 individus sont suffisants pour détecter une information basée sur un échiquier numérique de taille 512*512, soit en moyenne seulement 0.5 individu par cellule. Dans le cas catégoriel, l'échiquier de taille 512*512 se réduit à une grille 2*2 de 4 cellules après groupement des lignes et colonnes, et seulement 4000 individus sont nécessaires pour identifier correctement ce motif, pourtant "caché" dans un échiquier de 250000 cases.

4.2.4 Expérimentations complémentaires

Les résultats théoriques sur les seuils d'apprentissage présentés en section 4.1.4 sont basés sur l'utilisation de jeux de données particuliers, pour lesquels le critère d'évaluation d'un modèle en grille est calculable analytiquement.

On présente dans cette section une étude complémentaire basée sur des jeux de données artificiels générés aléatoirement selon des distributions connues, ce qui offre un éclairage probabiliste aux résultats théoriques sur les seuils d'apprentissage. Les résultats de cette section réutilisent des protocoles expérimentaux déjà présentés dans les annexes A et C, en les généralisant aux échantillons de très petites tailles et en étudiant l'impact de l'a priori de sélection de variables.

4.2.4.1 Résistance au bruit

On se donne une variable à expliquer booléenne équidistribuée et K variables explicatives numériques uniformément distribuées sur $[0, 1]$, indépendantes de la variable à expliquer. On s'intéresse d'abord au cas univarié ($K = 1$), en ignorant comme en annexe A l'a priori de sélection de variables , puis on étudie l'impact de l'a priori de sélection de variables lors du passage au multivarié.

Cas univarié sans a priori de sélection de variables. La variable explicative est évaluée au moyen d'une discrétisation par la méthode MODL. La discrétisation idéale est le modèle nul de discrétisation ne contenant qu'un seul intervalle, signifiant que la variable explicative n'est pas informative. En pratique, certains motifs peuvent apparaître aléatoirement, et on s'attend à ce qu'une certaine proportion des jeux de données donne lieu à des discrétisations multi-intervalles. L'objectif de cette expérimentation est d'évaluer cette proportion en fonction de la taille de l'échantillon.

Pour des tailles de 1 à 100000 individus, on génère aléatoirement un grand nombre (100000) d'échantillons, et on estime par comptage la proportion de modèles non nuls de discrétisation. La figure 4.3 affiche cette proportion pour chaque taille d'échantillon.

Les résultats confirment les seuils d'apprentissage pour la détection des discrétisations informatives. Il faut au minimum 6 individus pour détecter une discrétisation en deux in-

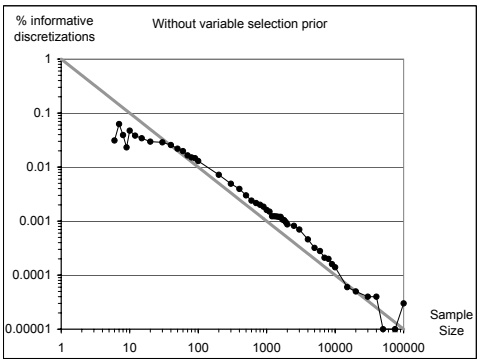


FIG. 4.3 – Proportion de discrétisations multi-intervalles pour une variable explicative numérique indépendante d'une variable booléenne à expliquer. La droite diagonale en grisé représente la fonction $f(N) = 1/N$.

tervalles, ce qui arrive dans environ 3% des cas "par hasard" (pour les deux discrétisations | 000 | 111 | et | 111 | 000 |, parmi les $64 = 2^6$ séquences possibles de booléens).

Pour 7 individus, $6\% \approx 8/128$ des discrétisations sont informatives : ce sont les discrétisations en deux intervalles purs, dont le premier intervalle comporte 2, 3, 4 ou 5 individus. Suivant le même schéma, on trouve $4\% \approx 10/256$ discrétisations informatives pour 8 individus, et $2.5\% \approx 12/512$ pour 9 individus. A partir de 10 individus, on voit apparaître des discrétisations comportant des intervalles non purs, comme par exemple | 010000 | 1111 |, pour un total de $5\% \approx 50/1024$ de discretisations informatives. A partir de 12 individus, il existe des discrétisations informatives à 3 intervalles, comme par exemple | 00 | 1111111 | 000 |.

Quand le nombre d'individus augmente, la proportion de discrétisations évaluées à tort comme informatives décroît très rapidement, approximativement en raison inverse de la taille de l'échantillon. Au delà de 10000 individus, la proportion de discrétisations informatives est très faible, ce qui explique le manque de fiabilité de son estimation en dépit des 100000 générations aléatoires de jeux d'essai.

En résumé, l'expérimentation confirme et quantifie le comportement aux limites de la méthode MODL dans le cas des échantillons de très petite taille. De plus, cette expérimentation montre que la méthode MODL est très résistante au bruit.

Résistance au bruit en multivarié. On s'intéresse cette fois à la résistance au bruit dans le cas de la discrétisation multivariée, afin d'évaluer l'impact de la prise en compte de l'aspect sélection de variables. La discrétisation multivariée idéale est le modèle nul de grille, ne comportant qu'une seule cellule.

Pour des tailles de 1 à 1000 individus, on génère aléatoirement un très grand nombre (un million) d'échantillons, et on estime par comptage la proportion de grilles contenant au moins deux cellules. La figure 4.4 affiche cette proportion pour chaque taille d'échantillon, pour $K = 1, 2, 10$, avec ou sans a priori de sélection de variable.

Les résultats montrent qu'en l'absence d'a priori de sélection de variables dans le critère d'évaluation d'une grille multivariée, la proportion de grilles détectées à tort comme

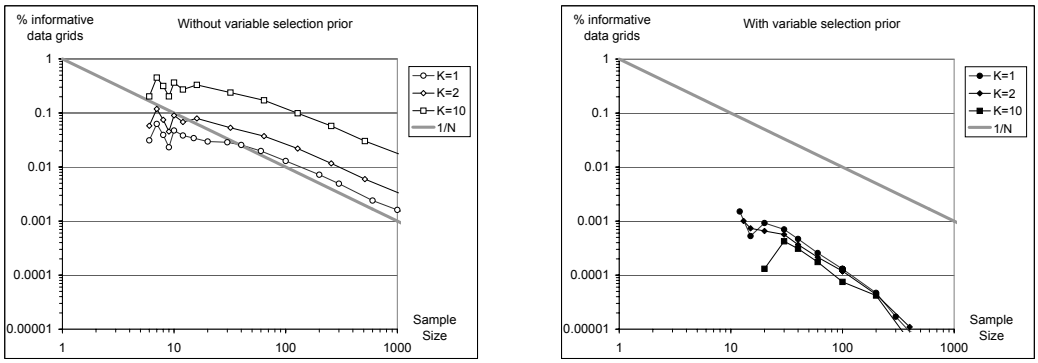


FIG. 4.4 – Proportion de grilles multi-cellules pour K de variables explicatives numériques indépendante d'une variable booléenne à expliquer, sans ou avec a priori de sélection de variable.

informatives augmente de façon quasi-linéaire avec le nombre de variables. Cela s'explique par le fait qu'un k-uplet de variables explicatives peut être identifié comme informatif au moyen d'un modèle en grille si l'une quelconque des K variables l'est au moyen d'une discrétisation univariée informative.

Quand on prend en compte l'a priori de sélection de variables, la proportion des discrétisations univariées ou multivariées informatives diminue très fortement, environ d'un facteur 100, et les courbes sont rapidement confondues pour les différents nombres de variables explicatives. L'a priori de sélection de variables renforce significativement la robustesse de la méthode, et la rend peu dépendante du nombre de variables de l'espace de représentation.

La meilleure résistance au bruit de la méthode est une conséquence directe de l'a priori de sélection de variables. En effet, les discrétisations univariée informatives doivent d'une part effectuer le choix de la variable informative (terme d'a priori $\log(K+1) + \log K$)), d'autre part choisir le nombre d'intervalles (terme d'a priori $\log N$), alors que les discrétisations non informatives se contentent de choisir un nombre nul de variables (terme $\log(K+1)$), le nombre d'intervalles étant automatiquement fixé à un.

En résumé, l'a priori de sélection de variables renforce très fortement la robustesse de la méthode et la rend indépendante du nombre de variables de l'espace de représentation. Ces améliorations se font au prix d'un seuil de détection des motifs informatifs légèrement plus élevé, nécessitant de l'ordre de $\log K + \log N$ individus supplémentaires.

4.2.4.2 Détection de motif complexe

On se donne une variable à expliquer booléenne équidistribuée et une variable explicative numérique uniformément distribuée sur $[0, 1]$, en dépendance avec la variable à expliquer selon un motif complexe. Ce motif est constitué d'une suite de 100 sous-séquences pures de fréquences égales, alternativement pour la valeur 0 et la valeur 1 de la variable à expliquer.

La variable explicative est évaluée au moyen d'une discrétisation par la méthode

MODL, en tenant compte de l'a priori de sélection de variables. La discrétisation idéale contient exactement 100 intervalles, mais il n'est pas possible de l'identifier s'il n'y a pas assez d'individus. Selon l'étude sur les seuils limites de détection des discrétisations informatives maximales, il suffit de 800 individus pour 100 intervalles de taille 8 exactement pour détecter le motif. Si cet échantillon théorique permet de mener à bien les calculs de façon analytique, il n'est pas représentatif dans la distribution des échantillons.

Dans cette expérience, pour des tailles de 1 à 100000 individus, on génère aléatoirement un grand nombre (100000) d'échantillons, et on mesure le nombre moyen d'intervalles des discrétisations apprises par la méthode MODL. La figure 4.5 représente pour chaque taille d'échantillon le nombre moyen d'intervalles.

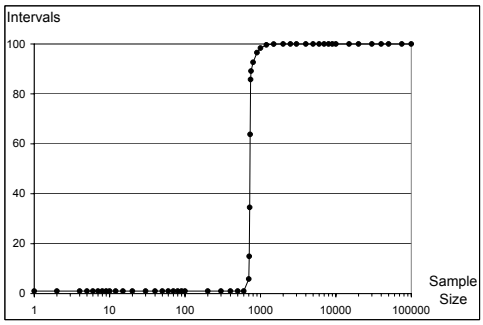


FIG. 4.5 – Nombre moyen d'intervalles I pour la discrétisation d'une variable explicative numérique en interaction complexe avec la variable booléenne à expliquer, suivant 100 séquences pures alternées.

Les résultats montrent un effet de seuil abrupt autour des échantillons de taille avoisinant 800 individus, confirmant et précisant ainsi les résultats de l'étude théorique sur les seuils d'apprentissage. Jusqu'à 680 individus, le nombre moyen d'intervalles est quasiment égal à 1. Il monte à environ 5 intervalles pour 700 individus et à 90 intervalles pour 750 individus. Au delà de 800 individus, le nombre moyen d'intervalles converge très rapidement vers exactement 100 intervalles, signifiant que le motif est correctement identifié.

Par rapport au critère ignorant l'aspect sélection de variables, les discrétisations informatives sont davantage pénalisées, et doivent exhiber des motifs informatifs comportant légèrement plus d'individus pour obtenir une probabilité a posteriori plus forte que celle du modèle nul. En refaisant l'expérimentation avec et sans a priori de sélection de variables, on constate que la détection du motif complexe nécessite une vingtaine d'individus supplémentaires quand l'a priori de sélection de variables est pris en compte. La figure 4.6 illustre la différence de comportement entre les deux a priori autour du seuil de détection de 800 individus.

En définitive, l'a priori de sélection de variables renforce très fortement, d'un facteur 100 environ, la fiabilité de la méthode, et la rend peu dépendante du nombre de variables explicatives. Cette fiabilité accrue ne nuit que faiblement à la finesse de la méthode, qui

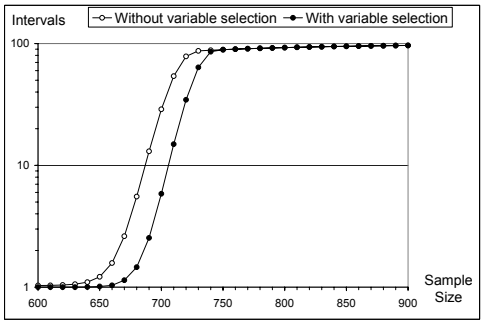


FIG. 4.6 – Impact de l'a priori de sélections de variables sur le seuil de détection d'un motif complexe de 100 séquence pures alternées.

nécessite de l'ordre quelques dizaines d'individus supplémentaires pour la détection de motifs complexes.

4.2.5 Bilan de l'évaluation des prétraitements

Les modèles en grille sont fondés sur un critère optimal au sens de Bayes et sur des algorithmes d'optimisation poussés, de complexité algorithmique super-linéaire par rapport au nombre d'individus. Une évaluation empirique intensive démontre que ces méthodes sont efficaces pour le prétraitement univarié et bivarié dans le cadre de la classification supervisée, en présentant notamment les caractéristiques suivantes :

- simplicité de mise en oeuvre :
  . aucun paramétrage utilisateur,
  . performance en temps de calcul,
  . capacité à gérer des échantillons de très petite comme de très grande taille,
- qualité de l'estimation :
  . approximateur universel de densité conditionnelle,
  . performance en taux de bonne prédiction,
  . reconnaissance de motif très fins,
  . robustesse et résistance au bruit,
- facilité d'interprétation :
  . qualité de l'estimation de l'importance prédictive par le taux de compression,
  . visualisation aisée de l'information prédictive au moyen d'histogrammes,
  . interprétation aisée de l'information prédictive par des règles de décision.

L'essentiel des résultats présentés est basé sur l'évaluation de l'apport intrinsèque des méthodes de prétraitement univarié et bivarié, considérées indépendamment de leur utilisation dans un classifieur. L'intégration de ces méthodes dans le classifieur Bayesien naïf est présentée en section 4.3, et leur extension au cas multivarié pour la classification supervisée et non supervisée est étudiée en section 4.4.

4.3 Améliorations du classifieur Bayesien naïf

Cette section expose de façon synthétique les travaux des articles [Boullé, 2005a, Boullé, 2006c, Boullé, 2007b] sur l'amélioration du classifieur Bayesien naïf.

Nous présentons plusieurs améliorations apportées au classifieur Bayesien naïf, notamment l'utilisation des méthodes MODL de discrétisation et groupement de valeurs pour bénéficier d'une estimation fiable et précise des densités conditionnelles univariées. Nous étudions ensuite l'intérêt des prétraitements bivariés pour contourner l'hypothèse naïve en étendant l'espace de représentation. Enfin, nous proposons de nouvelles méthodes pour la sélection de variables et le moyennage de modèles afin de tirer le meilleur parti des prétraitements MODL. L'ensemble de ces méthodes est évalué de façon comparative au moyen d'une expérimentation sur jeux de données réels.

4.3.1 Le classifieur Bayesien Naïf

Soient $X = (X_1, X_2, \ldots, X_K)$ un ensemble de K variables explicatives numériques ou catégorielles et Y une variable catégorielle à expliquer.

Le classifieur Bayesien associe à chaque individu la valeur à expliquer maximisant la probabilité conditionnelle $P(Y|X)$. Le classifieur Bayesien est optimal, mais il n'est pas calculable en pratique, puisqu'il suppose que l'on connaisse parfaitement la distribution de probabilité conditionnelle. Dans ce contexte, les modèles en grille pour la classification supervisée sont des approximateurs universels de densité conditionnelle, mais ils requièrent un nombre exponentiellement élevé d'individus vis à vis du nombre de variables explicatives (de l'ordre de 2^K selon le théorème 4.10), ce qui limite leur applicabilité dans le cas général. Ces modèles multivariés sont évaluée dans la section 4.4.

Le modèle Bayesien naïf relâche fortement la contrainte sur l'estimation multivariée de probabilité conditionnelle, en faisant l'hypothèse *naïve* d'indépendance des variables explicatives conditionnellement à la variable à expliquer. En appliquant cette hypothèse, on aboutit à :

$$P(Y|X) = P(Y|X_1, X_2, \ldots, X_K), \tag{4.11}$$

$$P(Y|X) = \frac{P(Y)P(X_1, X_2, \ldots, X_K|Y)}{P(X)}, \tag{4.12}$$

$$P(Y|X) = \frac{P(Y)\prod_{k=1}^{K} P(X_k|Y)}{P(X)}. \tag{4.13}$$

L'estimation de densité conditionnelle multivariée est alors factorisable sur les variables explicatives, ce qui permet de se ramener au problème plus simple de l'estimation de densité conditionnelle univariée.

Parfois qualifié d'*idiot de Bayes* dans la littérature, le classifieur Bayesien naïf est souvent performant en pratique sur de très nombreux jeux de données réels [Hand and Yu, 2001]. Il est simple à mettre en oeuvre, rapide à apprendre et à déployer, et n'est pas sujet au sur-apprentissage, puisque l'espace des modèles est réduit à un singleton. Du fait de la naïveté de son hypothèse et des redondances entre variables rencontrées en pratique, il conduit à une mauvaise estimation des probabilités conditionnelles multivariées. En

revanche, il est considéré comme performant pour l'ordonnancement de ces probabilités conditionnelles [Hand and Yu, 2001], ce qui lui permet d'être utilisable dans des applications de scoring, pour lesquelles les individus sont ordonnés par probabilité conditionnelle décroissante relativement à la valeur d'intérêt à expliquer.

4.3.2 Améliorations

Du fait de ses avantages, il est tentant d'améliorer le classifieur Bayesien naïf, en :
- utilisant un estimateur performant pour les probabilités conditionnelles univariées,
- contournant l'hypothèse naïve par construction de nouvelles variables capturant les interactions entre les variables explicatives initiales,
- sélectionnant un sous-ensemble de variables compatible avec l'hypothèse naïve,
- moyennant des modèles Bayesien naïfs issus de plusieurs sélections de variables, de façon à améliorer les performances prédictives en précision et fiabilité.

Ces pistes d'améliorations sont étudiées dans les sections suivantes.

4.3.2.1 Estimation conditionnelle univariée

Dans sa version originale [Langley *et al.*, 1992], la densité conditionnelle univariée est estimée par comptage pour les variables explicatives catégorielles et en faisant l'hypothèse d'une distribution Gaussienne par valeur à expliquer pour les variables explicatives numériques.

Dans le cas des variables catégorielles, l'estimation de la densité conditionnelle par comptage est justifiée pour des petits nombres de valeurs explicatives. L'estimation est en effet fiable sous cette contrainte, souvent vérifiée sur les jeux de données utilisés par la communauté scientifique (par exemple, pour les jeux de données UCI [Blake and Merz, 1996]). Dans la littérature, cette estimation par comptage est usuellement affinée au moyen de techniques de type lissage de Laplace. A notre connaissance, les méthodes de groupement de valeurs n'ont pas été exploitées dans le cas du classifieur Bayesien naïf.

En revanche, dans le cas numérique, l'estimation basée sur l'hypothèse Gaussienne a été largement discutée [Dougherty *et al.*, 1995, Liu *et al.*, 2002, Yang and Webb, 2002]. Les études montrent que la discrétisation des variables numériques, qui correspond à une estimation de densité conditionnelle non paramétrique constante par intervalle, permet d'améliorer de façon très significative les performances du classifieur Bayesien naïf. Parmi les méthodes de discrétisation étudiées, la méthode non supervisée de découpage en intervalles d'effectif égal obtient des performances compétitives, ce qui la rend intéressante du fait de sa simplicité et de son absence de risque de sur-apprentissage.

Dans les expérimentations, le classifieur Bayesien naïf noté NB(ref) utilise des estimations de densité conditionnelle par comptage dans le cas catégoriel, et au moyen d'une discrétisation non supervisée en 10 intervalles d'effectif égal dans le cas numérique. Ce classifieur est utilisé comme référence.

Les méthodes MODL de prétraitement univarié pour la classification supervisée produisent des estimations de densités conditionnelles à la fois fiables et précises. Chaque prétraitement, discrétisation ou groupement de valeurs, donne lieu à un tableau de contingence entre les intervalles (resp. groupes de valeurs) explicatifs et les valeurs à expliquer.

Ce tableau de contingence est utilisé pour estimer les probabilités conditionnelles par comptage.

Dans les expérimentations, le classifieur Bayesien naïf basé sur les méthodes MODL de prétraitement univarié est noté NB(MODL).

4.3.2.2 Estimation conditionnelle bivariée

Les méthodes MODL de prétraitement bivarié permettent d'estimer la densité de la variable à expliquer conditionnellement à une paire de variables explicatives considérées conjointement. Cela permet d'identifier avec deux variables des informations indétectables avec chaque variable prise isolément, comme dans le cas du XOR par exemple. Ces méthodes permettent donc de contourner partiellement l'hypothèse naïve.

Pour chaque paire de variables explicatives (X_1, X_2), on utilise le modèle en grille bivariée comme une méthode non paramétrique de construction d'une nouvelle variable explicative catégorielle $X_{(1,2)} = f(X_1, X_2)$, qui à chaque paire de valeurs explicatives associe l'index de sa cellule dans la grille bivariée. De façon similaire au cas univarié, les probabilités conditionnelles sont estimées par comptage localement à chaque cellule.

Dans le cas particulier où un individu en test aboutit selon $f(X_1, X_2)$ dans une cellule vide en apprentissage, il se voit associer un index spécial (au lieu d'un index de cellule), et une estimation de densité conditionnelle globale, par comptage sur l'ensemble de tous les individus de la grille.

On propose de prendre en compte les paires de variables de la façon la plus simple possible, en augmentant l'espace de représentation des K variables explicatives au moyen des $K(K-1)/2$ nouvelles variables construites à partir d'une analyse bivariée MODL exhaustive.

Dans les expérimentations, le classifieur Bayesien naïf basé sur les méthodes MODL de prétraitement bivarié est noté NB2(MODL).

4.3.2.3 Sélection de variables

La sélection de variables est utilisée pour plusieurs objectifs différents [Guyon and Elisseeff, 2003, Guyon *et al.*, 2006a], notamment pour l'amélioration des performances prédictives, la diminution du temps d'apprentissage et de déploiement, et une meilleure intelligibilité des modèles. Dans le cas du classifieur Bayesien naïf, l'intention première est l'amélioration des performances prédictives, en recherchant un sous-ensemble de variables aussi compatible que possible avec l'hypothèse naïve, donc avec un minimum de dépendance entre variables.

Les méthodes de sélection de variables peuvent se ranger en deux groupes : les méthodes filtres sont indépendantes du classifieur utilisé alors que les méthodes enveloppes [Kohavi and John, 1997] utilisent le classifieur comme une boîte noire afin d'évaluer chaque sous-ensemble de variables.

La sélection de variables pour le classifieur Bayesien naïf est appliquée dans le cadre de l'approche enveloppe en utilisant le taux de bonne prédiction (ACC=accuracy) comme mesure de la qualité du classifieur [Langley and Sage, 1994]. L'algorithme d'optimisation utilisé pour rechercher le meilleur sous-ensemble de variables est l'heuristique Forward

Feature Selection. Cette heuristique part d'un sous-ensemble de variables initialement vide, ajoute itérativement la variable apportant la meilleure amélioration selon la mesure de qualité utilisée, et continue tant qu'il y a amélioration.

Le taux de bonne prédiction est une mesure d'intérêt souvent limité en pratique [Provost *et al.*, 1998], notamment en cas de fort déséquilibre dans la distribution des valeurs à expliquer. De plus, le taux de bonne prédiction est une mesure de qualité peu discriminante : cette mesure prend au plus N valeurs distinctes alors que 2^K sous-ensembles de variables sont à évaluer, ce qui donne forcément lieu à beaucoup d'égalités. Dans [Boullé, 2006a], une amélioration du classifieur Bayesien naïf sélectif (SNB) est proposée en remplaçant le taux de bonne prédiction par l'aire sous la courbe de ROC (AUC = area under the ROC curve) [Fawcett, 2003]. Ainsi, on évalue l'ordonnancement des individus triés par probabilité conditionnelle décroissante pour chaque valeur à expliquer, ce qui permet de tenir compte de la distribution complète à expliquer au lieu de la valeur à expliquer majoritaire uniquement.

La sélection de variables (basée sur le critère ACC ou AUC) rencontre ses limites sur des jeux de données comportant de très grands nombres de variables. En mesurant l'apport marginal de chaque nouvelle variable dans la sélection, on observe que les premières variables ajoutées dans la sélection ont un apport significatif, alors que les dernières ont un apport négligeable, pratiquement indiscernable du bruit. Cet effet est de faible importance si seule la performance prédictive est visée. Il est en revanche fortement dommageable quand l'intelligibilité du modèle est recherchée.

Dans [Boullé, 2006c], la sélection de variables est reformulée comme un problème de sélection de modèles, et résolue selon une approche Bayesienne. On considère que l'on dispose d'un ensemble d'estimateurs de densité conditionnelle élémentaires, issus empiriquement des prétraitements MODL.

La méthode évalue alors chaque hypothèse de sélection de variables, en utilisant un a priori uniforme sur le nombre K_s de variables sélectionnées, puis sur le choix du sous ensemble des K_s variables choisies parmi K. Cet a priori est identique à celui utilisé dans le cas des modèles en grille multivariés (cf. formule 2.12). Une fois K_s variables choisies, la formule de Bayes permet un calcul exact de la probabilité conditionnelle des valeurs à expliquer. Cette probabilité conditionnelle est à la base de l'estimation de la vraisemblance conditionnelle des modèles de sélection de variables dans le cas du classifieur Bayesien naïf.

Un nouvel algorithme de recherche du meilleur modèle de sélection est également proposé en remplacement de l'heuristique usuelle de Forward Feature Selection. Cet algorithme alterne des passes d'ajouts (Forward) et de retraits (Backward) de variables par parcours simples des variables, réordonnées aléatoirement entre chaque passe. Ce processus, qui converge très vite, est répété plusieurs fois de façon à améliorer la performance tout en diminuant la variance du résultat de sélection. La complexité algorithmique globale de cette heuristique est $O(KN \log(KN))$.

Les trois méthodes de sélection de variables présentées ci-dessus sont basées respectivement sur l'optimisation du taux de bonne prédiction (SNB(ACC)), de l'aire sous la courbe de ROC (SNB(AUC)), ou de la probabilité a posteriori de la sélection (SNB(MAP)). Elles sont évaluées dans [Boullé, 2006c], sur trois critères : taux de bonne prédiction, aire sous la courbe de ROC et log vraisemblance conditionnelle négative (ce qui correspond au critère

ILF=Information Loss Function [Witten and Frank, 2000]). Aucune des trois méthodes ne domine les deux autres, chacune étant la meilleure pour le critère d'évaluation correspondant au critère de sélection de variables qu'elle optimise. Néanmoins, la méthode SNB(MAP) fondée sur l'approche Bayesienne n'est que légèrement dominée sur les critères ACC et AUC, alors qu'elle domine très nettement les deux autres méthodes sur le critère ILF. Elle conduit à des sous-ensembles de variables de taille faible, ce qui en améliore grandement l'interprétabilité.

Dans les expérimentations, le classifieur Bayesien naïf sélectif retenu est celui fondé sur l'approche Bayesienne de la sélection de variables. Il est noté SNB(MAP) dans le cas univarié et SNB2(MAP) dans le cas bivarié.

4.3.2.4 Moyennage de modèles

Le moyennage de modèles vise à combiner la prédiction d'un ensemble de classifieurs de façon à améliorer les performances prédictives.

Ce principe a été appliqué avec succès dans le cas du bagging [Breiman, 1996], qui exploite un ensemble de classifieurs appris sur une série de sous-échantillons d'apprentissage obtenus par ré-échantillonnage (boosting) de l'échantillon initial. Dans le cas des arbres de décision [Breiman *et al.*, 1984], la méthode a été étendue aux Random Forests [Breiman, 2001], en échantillonnant non seulement l'espace des individus, mais également l'espace des variables de façon à augmenter la diversité des classifieurs moyennés. Dans ces approches, le classifieur moyenné résultant procède par vote des classifieurs élémentaires pour effectuer sa prédiction.

A l'opposé des approches de type bagging, où chaque classifieur élémentaire se voit attribuer le même poids, le moyennage Bayesien de modèles (BMA= Bayesian Model Averaging) [Hoeting *et al.*, 1999] pondère les classifieurs selon leur probabilité a posteriori. L'approche BMA a donné lieu à des résultats théoriques d'optimalité attractifs, sous condition que l'ensemble des jeux de données "de l'univers" suive la distribution a priori des modèles envisagés [Raftery and Zheng, 2003]. Elle est par contre souvent difficile à mettre en oeuvre en pratique, en requérant l'évaluation de la probabilité a posteriori des modèles et une méthode d'échantillonnage de la distribution a posteriori des modèles.

L'approche BMA est appliquée pour moyenner les sélections de variables effectuées dans le classifieur Bayesien naïf sélectif SNB(MAP) [Boullé, 2006c]. Pour ce classifieur, la probabilité a posteriori d'un modèle est calculable en $O(KN)$, par parcours de l'ensemble des N individus de l'échantillon d'apprentissage et estimation de leur probabilité a posteriori au moyen de la formule du classifieur Bayesien naïf comportant K termes. Il est à noter que dans le cas des méthodes Forward ou Backward d'optimisation de la sélection de variables, la probabilité a posteriori des modèles peut être recalculée en $O(N)$ suite à chaque ajout ou retrait de variables, puisqu'un seul terme est modifié dans la formule du classifieur Bayesien naïf.

De façon pragmatique, on échantillonne la distribution des modèles en collectant l'ensemble des modèles évalués lors de l'heuristique d'optimisation de la sélection de variables. La méthode résultante, baptisée SNB(BMA), conserve ainsi la même complexité algorithmique que la méthode SNB(MAP).

La méthode SNB(BMA) moyenne un grand nombre de classifieurs Bayesiens naïfs issus de multiples sélections de variables, chacun des classifieurs moyennés pouvant s'interpréter comme un moyennage de probabilités conditionnelles univariées sur les variables sélectionnées. Ce double moyennage permet de transposer le moyennage sur les modèles à un moyennage sur les variables. On obtient alors une variante de la formule de prédiction du classifieur Bayesien naïf, où chaque variable se voit attribuer un coefficient de pondération compris entre 0 et 1. Il s'agit d'une sélection "douce" des variables, qui permet au classifieur Bayesien naïf sélectif moyenné de rester proche conceptuellement d'un classifieur Bayesien naïf sélectif élémentaire. Cette proximité conceptuelle facilite à la fois le déploiement des modèles et leur intelligibilité, ce qui est rarement le cas des modèles moyennés.

En dépit de ces avantages théoriques et pratiques, les expérimentations présentées dans [Boullé, 2006c] montrent que la méthode SNB(BMA) n'apporte pas d'amélioration notable par rapport à la méthode SNB(MAP) en termes de performance prédictive. En inspectant la distribution a posteriori des modèles (de façon exhaustive quand $K \leq 20$), on s'aperçoit que celle-ci est extrêmement piquée, si bien que le moyennage des modèles se réduit quasiment au modèle MAP. Afin de trouver un compromis entre des poids identiques pour les modèles comme dans le bagging, et une distribution des poids très fortement déséquilibrée comme dans le moyennage BMA, une approche intermédiaire heuristique est proposée dans [Boullé, 2006c], en utilisant un lissage logarithmique de la distribution a posteriori des modèles. Ce schéma de pondération, baptisée Compression based Model Averaging (CMA) revient à pondérer les modèles par leur taux de compression (cf. section 4.1.2).

Le classifieur Bayesien naïf sélectif moyenné résultant est étudié et expérimenté de façon comparative sur de nombreuses bases réelles dans [Boullé, 2006c]. Les résultats montrent que cette méthode domine très nettement et très significativement les méthodes alternatives de sélection ou moyennage de modèle sur tous les critères d'évaluation : taux de bonne prédiction, aire sous la courbe de ROC et log vraisemblance conditionnelle négative.

Dans les expérimentations, le classifieur Bayesien naïf sélectif moyenné retenu est celui fondé sur la pondération des modèles par leur taux de compression. Il est noté SNB(CMA) dans le cas univarié et SNB2(CMA) dans le cas bivarié.

4.3.3 Protocole d'évaluation

De façon à évaluer l'apport respectif de chaque amélioration du prédicteur Bayesien naïf, on procède à une expérimentation sur 30 jeux de données réels, issus de l'UCI [Blake and Merz, 1996]. Ces jeux de données, décrits de façon synthétique dans le tableau 3 de l'annexe C, représentent une large variété de cas, avec des variables explicatives numériques et catégorielles, des tailles comprises entre 150 et 50000 individus pour 4 à 60 variables explicatives, et des valeurs à expliquer en nombre compris entre 2 et 28.

Les différentes versions du classifieur Bayesien naïf retenue pour l'évaluation sont :

- NB(ref) : version de référence, sans prétraitement dans le cas catégoriel avec une discrétisation non supervisée en 10 intervalles d'effectif égal,

– NB(MODL) : prétraitements MODL univariés,
– NB2(MODL) : prétraitements MODL bivariés,
– SNB(MAP) : sélection de variables suivant une approche Bayesienne, en univarié,
– SNB2(MAP) : sélection de variables suivant une approche Bayesienne, en bivarié,
– SNB(CMA) : moyennage de modèle par taux de compression, en univarié,
– SNB2(CMA) : moyennage de modèle par taux de compression, en bivarié.

Les critères d'évaluation étudiés sont :
– ACC : taux de bonne prédiction,
– AUC : aire sous la courbe de ROC,
– ILF : log vraisemblance conditionnelle négative.

Les critères ACC et AUC prennent leurs valeurs entre 0 et 1. On choisit de normaliser également l'ILF selon la formule $(1 - ILF/ILF_\emptyset)$ où $ILF_\emptyset$ représente l'ILF du modèle Bayesien naïf ne prenant en compte aucune variable explicative.

Chaque critère d'évaluation est mesuré lors d'une validation croisée stratifiée à 10 niveaux, sur les 30 jeux de données de l'expérimentation.

4.3.4 Résultats d'évaluation

L'objectif de cette expérimentation est de présenter de façon synthétique les évaluations menées de façon parcellaire dans plusieurs articles (cf. table 4.1), se focalisant respectivement sur l'aspect groupement de valeurs univarié, sélection de variables et moyennage de modèles, et prétraitements bivariés.

Tab. 4.1 – Evaluation des apports au classifieur Bayesien naïf par des expérimentations sur des bases réelles.

Objet de l'évaluation	Méthodes	Critères	Jeux de données
Groupement de valeurs [Boullé, 2005a]	NB(ref) NB(MODL) 5 méthodes alternatives	ACC	12 jeux de données de l'UCI (uniquement variables catégorielles)
Sélection de variables Moyennage de modèles [Boullé, 2006c]	NB(ref) NB(MODL) SNB(ACC) SNB(AUC) SNB(MAP) SNB(BMA) SNB(CMA)	ACC AUC ILF	30 jeux de données de l'UCI 10 jeux de données de challenges
Prétraitements bivariés [Boullé, 2007b]	NB(MODL) NB2(MODL) SNB(CMA) SNB2(CMA)	ACC	30 jeux de données de l'UCI

Les résultats détaillés de ces évaluations montrent que les différences entre les méthodes

sont significatives. Trois indicateurs de performance synthétiques, valeur moyenne, rang moyen et nombre de différences significatives, produisent un résumé consistant des performances et un ordonnancement global quasiment identique des méthodes par critère d'évaluation. On utilise ici la valeur moyenne de chaque critère sur l'ensemble des 30 jeux de données comme indicateur synthétique de performance des méthodes évaluées.

La figure 4.7 présente les résultats d'évaluation synthétiques sur deux plans bicritères AUC*ACC et AUC*ILF.

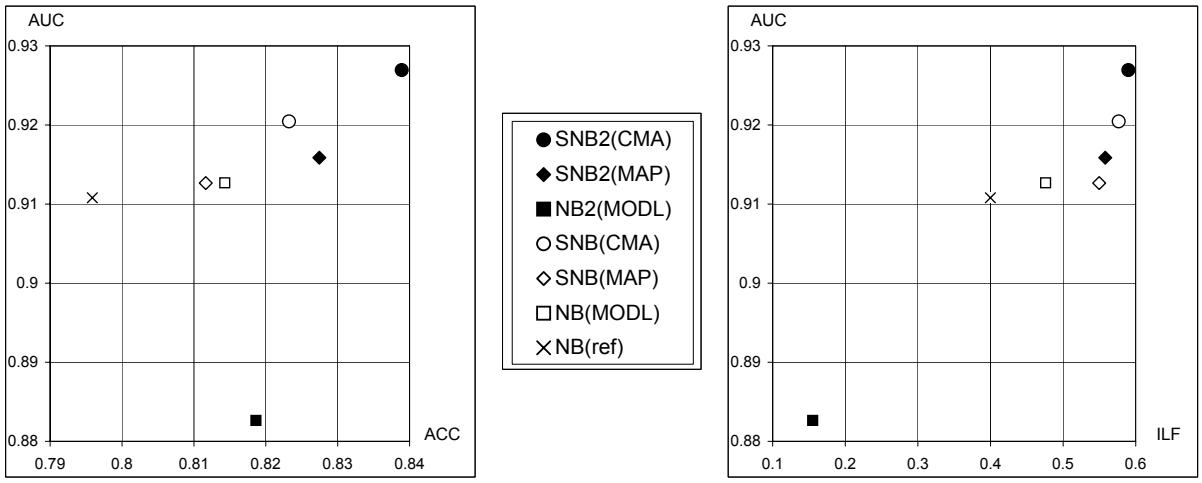


FIG. 4.7 – Moyenne des critères ACC, AUC et ILF pour l'évaluation de classifieurs Bayesien naïfs sur 30 jeux de données de l'UCI.

Apport des prétraitements univariés. La méthode NB(MODL) domine la méthode NB(ref) sur l'ensemble des trois critères d'évaluation, ce qui démontre l'apport des prétraitements univariés MODL. De plus, la robustesse des méthodes MODL leur permet d'éliminer les variables explicatives inutiles, indépendantes de la variable à expliquer.

Apport de la sélection de variables. La méthode SNB(MAP) obtient des résultats comparables à ceux de la méthode NB(MODL) sur les deux critères ACC et AUC, et des résultats significativement meilleurs sur le critère ILF.

En recherchant un sous-ensemble de variables explicatives compatible avec l'hypothèse naïve, la méthode SNB(MAP) sélectionne un petit nombre de variables : on passe de 20 variables en moyenne sur les jeux de données à 5.5 variables seulement après sélection. Les variables redondantes sont ainsi éliminées, ce qui améliore fortement la qualité de l'estimation de densité conditionnelle mesurée par le critère ILF. L'intelligibilité des modèles est également améliorée, puisque le faible nombre de variables sélectionnées s'interprète comme un sous-ensemble de variables conditionnellement indépendantes.

Apport du moyennage de modèles. La méthode SNB(CMA) domine la méthode SNB(MAP) sur l'ensemble des trois critères d'évaluation.

En pondérant les modèles par leur taux de compression, la méthode SNB(CMA) exploite l'ensemble de l'espace de représentation en combinant efficacement chaque sélection de variables. L'ensemble des variables ayant un poids non négligeable est d'environ 15 en moyenne, à comparer avec les 20 variables de la représentation initiale et les 6 du modèle SNB(MAP). En inspectant finement les poids des variables, on observe que la somme des poids des variables est approximativement la même pour les modèles SNB(CMA) et SNB(MAP). Tout se passe comme si le poids de chaque variable dans SNB(MAP) était distribué sur les variables redondantes dans SNB(CMA). Cet effet contribue à une amélioration des performances et de la robustesse du classifieur moyenné.

Apport des prétraitements bivariés seuls. L'utilisation naïve des prétraitements bivariés dans la méthode NB2(MODL) n'améliore que faiblement le critère ACC et détériore très fortement les critères AUC et ILF par rapport au classifieur univarié NB(MODL).

Quand on passe au bivarié, deux effets entrent en concurrence suite à la construction des nouvelles variables. Les interactions entre variables, capturées dans les grilles MODL bivariées, permettent d'enrichir l'espace de représentation et de détecter de nouvelles informations masquées en univarié. A l'opposé, la construction d'un grand nombre de nouvelles variables augmente fortement les redondances, incompatibles avec l'hypothèse naïve. Une inspection détaillée des résultats en annexe C montre que ces deux effets sont observés, avec de fortes améliorations ou dégradations du taux de bonne prédiction sur une partie importante des jeux de données. L'apport de l'analyse bivariée est alors négligeable en moyenne pour le taux de bonne prédiction, et il entraîne une variance accrue des résultats. Sur les deux autres critères, qui mesurent plus (ILF) ou moins (AUC) finement la qualité des estimations de probabilité conditionnelle, la dégradation des performances est très forte (ILF) à forte (AUC) en raison des très nombreuses redondances de l'espace de représentation. Globalement, l'impact du bivarié dans le prédicteur Bayesien naïf est fortement négatif.

Apport des prétraitements bivariés avec sélection de variables et moyennage de modèles. L'utilisation des prétraitements bivariés de façon conjuguée avec la sélection de variables dans SNB2(MAP) et surtout avec le moyennage de modèles dans SNB2(CMA) améliore très fortement et significativement les performances sur les critères ACC et AUC, un peu moins nettement sur le critère ILF. Une inspection détaillée des résultats sur le critère ACC montre que la méthode SNB2(CMA) apporte une amélioration moyenne de 1.5% par rapport à SNB(CMA), significative dans la moitié des cas. Les performances en ACC ne sont jamais dégradées significativement, et sont améliorées de 3% à 15% sur un quart des jeux de données.

En résumé, les prétraitements bivariés sont inutiles voire nuisibles dans le cas du classifieur Bayesien naïf s'ils sont utilisés seuls. Accompagnés d'une méthode de sélection et moyennage de modèle, ils permettent de contourner l'hypothèse naïve de façon effective, ce qui se traduit par une amélioration significative des performances prédictives.

4.3.5 Bilan des améliorations du classifieur Bayesien naïf

Cette section récapitule les apports des méthodes présentées, puis en présente les axes d'amélioration.

4.3.5.1 Apports des méthodes présentées

Dans cette section, nous avons présenté une série d'améliorations du classifieur Bayesien naïf. Chaque amélioration a un impact positif sur les performances prédictives, et les apports se cumulent. Le classifieur Bayesien naïf, souvent considéré comme performant, bénéficie ainsi d'un accroissement important de ses capacités prédictives, notamment pour l'estimation des probabilités conditionnelles.

On récapitule ci-dessous les principales caractéristiques de chaque amélioration.

- Prétraitement MODL univarié :
  . amélioration des performances prédictives,
  . suppression des variables inutiles,
  . complexité algorithmique de $O(KN \log N)$ en apprentissage et $O(K)$ par individu en déploiement.
- Sélection de variables selon une approche Bayesienne :
  . amélioration des performances prédictives,
  . suppression des variables redondantes,
  . complexité algorithmique de $O(KN \log(KN))$ en apprentissage et $O(K_s)$ par individu en déploiement,
  . intelligibilité accrue du modèle suite à la sélection de variables faiblement redondantes.
- Moyennage de modèles selon une pondération basée sur le taux de compression :
  . amélioration des performances prédictives,
  . amélioration de la robustesse grâce au moyennage sur la distribution des modèles,
  . complexité algorithmique de $O(KN \log(KN))$ en apprentissage et $O(K)$ par individu en déploiement,
  . intelligibilité dégradée du modèle, due aux poids sur les variables.
- Prétraitement MODL bivarié, avec sélection de variables et moyennage de modèles :
  . amélioration des performances prédictives (si prétraitements bivariés utilisés en conjonction avec la sélection de variables et le moyennage de modèles),
  . complexité algorithmique de $O(K^2N \log(KN))$ en apprentissage et $O(K^2)$ par individu en déploiement.
  . intelligibilité faible du modèle.

L'analyse des caractéristiques de chaque amélioration montre que les apports en performance prédictive se paient par une dégradation des temps d'apprentissage et de déploiement de modèles, avec des impacts également sur l'intelligibilité des modèles. En utilisant une analyse multi-critères, toutes les variantes présentées du classifieur Bayesien naïf sont optimale au sens de Pareto. Cela signifie qu'une amélioration sur un des critères (les performances prédictives par exemple) entraîne une détérioration sur un autre critère (temps d'apprentissage ou intelligibilité par exemple). Seul le contexte et les contraintes applicatives permettent de choisir une des variantes du classifieur Bayesien naïf.

4.3.5.2 Axes d'amélioration des méthodes présentées

Les méthodes présentées résultent de compromis pragmatiques, qui, s'ils ont prouvé leur efficacité, n'en sont pas moins questionnables.

1. La méthode de sélection de variables fait l'hypothèse de la disponibilité d'un ensemble d'estimateurs de densité conditionnelle élémentaires parfaits, et se focalise uniquement sur le choix d'un sous-ensemble de ces estimateurs. Cette hypothèse est fondée en pratique sur la fiabilité et la précision des méthodes MODL de prétraitement. De façon théorique, il faudrait également tenir compte de l'incertitude sur la qualité des estimateurs.
2. La méthode de moyennage de modèles résulte d'un compromis heuristique entre des poids de modèles identiques comme dans le moyennage de type bagging et des poids basés sur la distribution a posteriori des modèles. Cette option, très performante en pratique, n'est pas justifiée théoriquement.
3. L'algorithme d'échantillonnage des modèles utilisés pour le moyennage se contente de collecter les modèles évalués lors de l'optimisation MAP des modèles. Ce choix est très efficace d'un point de vue algorithmique, puisque le temps d'apprentissage du modèle moyenné est le même que celui de l'optimisation de la sélection de variables. En revanche, la qualité de l'échantillonnage n'est pas assurée.
4. Les prétraitements bivariés apportent un bénéfice certain en performance prédictive, mais ils entraînent un accroissement de la complexité algorithmique, quadratique avec le nombre de variables. Pour être applicable aux jeux de données comprenant des grands nombres de variables, l'analyse bivariée exhaustive doit être remplacée par une stratégie d'échantillonnage des paires de variables.
5. Les prétraitements bivariés sont utilisés de façon basique, puisque les nouvelles variables construites à partir des paires de variables initiales sont exploitées de façon agnostique, en oubliant leur origine. Il serait intéressant par exemple de considérer pour chaque paire de variables l'alternative entre une estimation de densité conditionnelle factorisée sur les estimations univariées et une estimation bivariée évaluée directement par un modèle en grille bidimensionnelle.

4.3.5.3 Hypothèse Bayesienne naïve et hypothèse Bayesienne ingénue

Les méthodes présentées pour améliorer le classifieur Bayesien naïf ont démontré leur apport significatif en performances prédictives lors d'une expérimentation sur 30 jeux de données de l'UCI. Des expérimentations portant sur des jeux de données plus complexes, comportant de grands nombres de variables, sont présentés en section 4.5.

Il est à noter que le classifieur Bayesien naïf résout le problème de l'estimation de densité conditionnelle sur l'ensemble de toutes les variables explicatives en factorisant cette estimation sur chacune des variables grâce à l'hypothèse *naïve* d'indépendance conditionnelle.

A l'inverse, les modèles en grille s'attaquent à l'estimation de densité conditionnelle directement sur toutes les variables conjointement. On pourrait qualifier cette approche d'*ingénue*, tant elle paraît irréaliste. Cette approche est évaluée en section 4.4.

4.4 Evaluation des modèles en grille multivariés

Les modèles en grille de données partitionnent chaque variable numérique en intervalles et chaque variable catégorielle en groupes de valeurs. Le produit cartésien de ces partitions univariées définit une partition multivariée de l'ensemble des individus, permettant de construire un estimateur de densité non paramétrique.

On introduit dans cette section plusieurs méthodes d'apprentissage exploitant les modèles en grille dans le cadre de la classification supervisée, de la classification non supervisée et du coclustering des individus et des variables. Ces méthodes sont évaluées au moyen d'une expérimentation sur jeux de données réels.

4.4.1 Classification supervisée

On présente deux méthodes pour construire un classifieur en grille à partir des modèles en grille, que l'on évalue ensuite sur des jeux de données réels.

4.4.1.1 Classification par grille

Le critère d'évaluation utilisé est celui des modèles en grille pour la classification supervisée, définis en section 2.1 (formule multivariée avec sélection de variables 2.13). La méthode d'optimisation, présentée en section 3.2, comprend une heuristique gloutonne ascendante, des algorithmes de post-optimisation et une méta-heuristique d'exploration de l'espace des modèles. On retient ici le meilleur modèle issu de l'optimisation pour construire un classifieur.

Pour chaque individu en test, on recherche la cellule de la grille d'apprentissage déterminée par les valeurs explicatives de l'individu. Cette cellule permet d'associer à l'individu une distribution des valeurs à expliquer, estimée par comptage sur la cellule apprise, et une prédiction, correspondant à la valeur à expliquer majoritaire sur la cellule. Pour les individus en test aboutissant dans une cellule vide, on associe la distribution et la prédiction globale par comptage sur l'ensemble de la grille.

Dans les expérimentations, le classifieur par grille de données (Data Grid) est baptisé DG(MAP).

4.4.1.2 Classification par moyennage de grilles

L'étude sur les seuils d'apprentissage montre que les grilles de données sont limitées en dimension à environ $\log_2 N$ variables explicatives. Il est alors naturel d'envisager de combiner plusieurs grilles pour améliorer les performances en prédiction.

Etant donné que les probabilités a posteriori des modèles en grille sont disponibles de façon exacte (cf. chapitre 2), le moyennage Bayesien BMA [Hoeting *et al.*, 1999] paraît indiqué. Il faut néanmoins recourir à une méthode d'échantillonnage de la distribution a posteriori des modèles, par exemple au moyen de méthodes MCMC [Neal, 1993, Robert, 2006]. Précisons préalablement quelques contraintes liées aux modèles en grille :

1. l'espace des modèles est discret, organisé suivant cinq niveaux de hiérarchie (nombre de variables, sous-ensemble de variables, tailles des partitions univariées, partitions

univariées en groupes ou intervalles, distributions des valeurs à expliquer par cellule explicative),

2. la taille de l'espace des modèles est très importante, avec de l'ordre de $(2^N)^K$ paramètres au dernier niveau de la hiérarchie,
3. chaque modèle peut contenir de l'ordre de N paramètres pour une grille de dimension $\log_2 N$, ce qui implique que les paramètres d'un sous-ensemble de seulement K modèles peuvent être aussi coûteux en stockage mémoire que le tableau complet individus * variables des données en apprentissage,
4. les modèles en grille sont difficiles à optimiser ; les échantillonner correctement paraît plus que difficile,
5. la distribution a posteriori des modèles en grille semble extrêmement piquée (cf. section 4.4.1.3), ce qui rend le moyennage inopérant, puisque le modèle moyenné est quasi identique au modèle MAP.

Avant d'affronter directement les difficultés prévisibles, et afin d'explorer les apports potentiels du moyennage de modèles, on fait le choix pragmatique d'une approche similaire à celle utilisée avec succès pour le moyennage de classifieurs Bayesiens naïfs.

On échantillonne ainsi la distribution des modèles en grille en collectant dans un ensemble $\mathbb{M}$ tous les optima locaux issus de la méta-heuristique d'optimisation (les doublons sont éliminés). Les modèles M de $\mathbb{M}$ sont pondérés par leur taux de compression $g(M)$. Le modèle moyenné est alors utilisé pour prédire la probabilité conditionnelle $P(Y|X)$ selon la formule heuristique :

$$P(Y|X,\mathbb{M}) = \frac{1}{\sum_{M\in\mathbb{M}} g(M)} \sum_{M\in\mathbb{M}} g(M)P(Y|X,M). \tag{4.14}$$

Dans les expérimentations, le classifieur par moyennage de grilles résultant de cette approche préliminaire fortement heuristique est baptisé DG(CMA).

4.4.1.3 Illustration sur la base Waveform

La base Wavefom [Breiman *et al.*, 1984] comporte 21 variable explicatives numériques et 5000 individus. La variable à expliquer est équidistribuée sur trois valeurs.

On utilise 70% des individus pour l'apprentissage et 30% en test. Dans cette expérimentation, on collecte tous les optimaux locaux des modèles en grille obtenus par la méta-heuristique d'optimisation, sans se contraindre en temps de calcul.

La figure 4.8 trace les taux de compression des optimaux locaux obtenus dans les 1000 premières secondes d'apprentissage (sur un PC 3.2 Ghz). Environ 1600 modèles optimaux locaux sont obtenus, soit en moyenne moins de une seconde par modèle. Le premier optimum local est obtenu dès la première seconde, et n'est dépassé que cinq fois parmi les 1600 modèles. La première amélioration est obtenue au bout de 240 secondes, avec un modèle environ 10000 fois plus probable que le premier optimum local. Le modèle ayant la meilleur probabilité a posteriori est obtenu au bout de 980 secondes, et est 50000 fois plus probable que le premier optimum local.

La figure 4.8 montre en définitive que du point de vue du taux de compression, l'heuristique d'optimisation est efficace et permet de trouver rapidement un bon modèle. Du

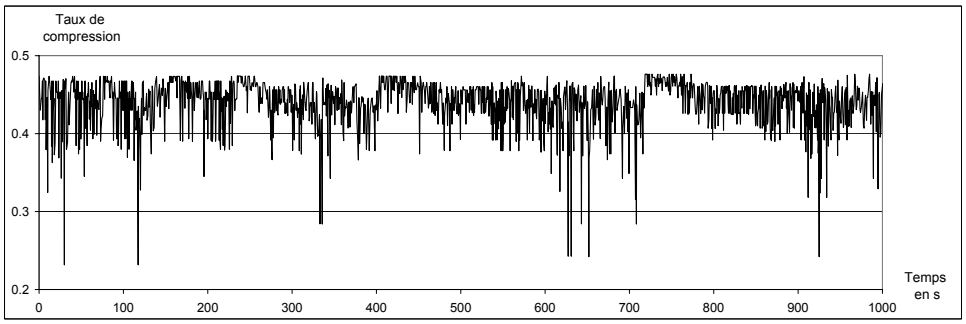


FIG. 4.8 – Optimisation des modèles en grille sur le jeux de données Waveform. Le taux de compression des modèles obtenus comme optimaux locaux de la méta-heuristique d'optimisation est tracé en fonction du temps.

point de vue de la vraisemblance conditionnelle (exponentielle par rapport au taux de compression), l'efficacité est nettement moindre, puisqu'il faut un temps très important pour obtenir des améliorations, et que chaque nouveau modèle MAP empirique est fortement plus probable que le MAP précédent. Dans ce contexte, tout moyennage Bayesien se réduit au modèle MAP empirique.

La figure 4.9 représente la fonction de répartition des probabilités a posteriori des modèles en grille, reconstituée empiriquement à partir des optimaux locaux de la figure 4.8. Le moyennage DG(CMA) combine l'ensemble des modèles en grille obtenus comme optimaux locaux de la méta-heuristique d'optimisation, en les pondérant par leur taux de compression.

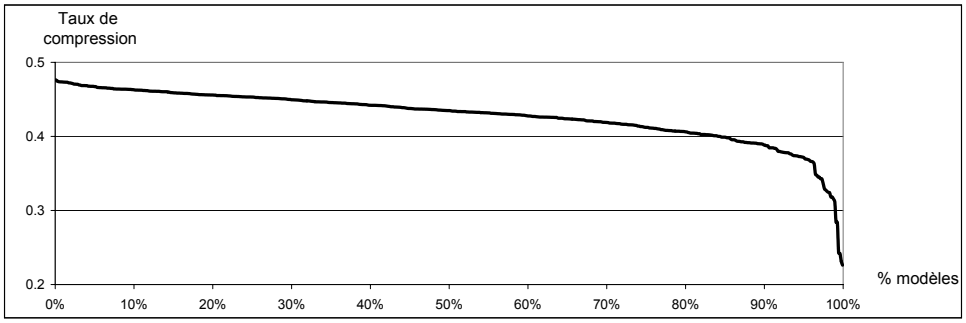


FIG. 4.9 – Fonction de répartition empirique des probabilités a posteriori des modèles en grille sur le jeu de données Waveform, représentée sur une échelle logarithmique par leur taux de compression. Par exemple, les 10% des modèles les plus probables sur la gauche de la figure ont tous un taux de compression supérieur à 0.45.

En évaluant le modèle DG(MAP) en test, on obtient des performances de 0.73 en ACC, 0.90 en AUC et 0.48 en ILF. Le modèle moyenné DG(CMA), qui combine des centaines de modèles, améliore significativement les performances avec 0.81 en ACC, 0.94 en AUC et 0.54 en ILF. Ce résultat préliminaire illustre le mécanisme du moyennage de modèles par taux de compression ainsi que son potentiel d'amélioration des performances prédictives.

4.4.1.4 Evaluation de l'apport prédictif

Le protocole d'évaluation utilisé est le même que dans le cas du classifieur Bayesien naïf de la section 4.3, avec une validation croisée stratifiée à 10 niveaux pour l'évaluation des critères ACC, AUC et ILF sur les 30 jeux de données de l'UCI décrits dans le tableau 3 de l'annexe C.

A titre de référence, on reprend certains résultats représentatifs de l'évaluation des classifieurs Bayesien naïfs. Les classifieurs participant à l'évaluation des modèles de classification en grille sont ici :

- classifieur en grille :
  - . DG(MAP) : utilisation du meilleur modèle en grille,
  - . DG(CMA) : moyennage de modèles en grille par taux de compression,
- classifieur Bayesien naïf :
  - . NB(MODL) : prétraitements MODL univariés,
  - . SNB(MAP) : utilisation de la meilleure sélection de variables,
  - . SNB(CMA) : moyennage des sélections de variables par taux de compression.

L'algorithme d'optimisation des modèles en grille est paramétré de façon à produire une centaine d'optima locaux dans la méta-heuristique, ce qui correspond à un temps d'apprentissage environ dix fois supérieur à celui de classifieur Bayesien naïf sélectif.

Comme dans l'évaluation des classifieurs Bayesiens naïfs, on utilise ici la valeur moyenne de chaque critère sur l'ensemble des 30 jeux de données comme indicateur synthétique de performance des méthodes évaluées. La figure 4.10 présente les résultats d'évaluation synthétiques sur deux plans bicritères AUC*ACC et AUC*ILF. Les modèles en grille et les modèles Bayesiens naïfs ont des biais radicalement opposés. Leurs performances respectives varient beaucoup, avec des différences de -20% à $+20\%$ selon les jeux de données. Les indicateurs de performance macroscopiques reflètent mal la diversité des résultats détaillés. Contrairement à l'évaluation des classifieurs Bayesiens naïfs de la section 4.3, il n'y a pas de relation de domination quasi-systématique, ni entre modèles en grille et modèles Bayesiens naïfs, ni entre modèles en grille simples et modèles en grille moyennés pour le critère ILF. Il est nécessaire de procéder à une analyse plus fine, jeu de données par jeu de données.

Apport des modèles en grille. La figure 4.11 présente les différences par jeux de données pour le critère ACC, en prenant pour référence la performance du classifieur Bayesien naïf NB(MODL). Sur les jeux de données pour lesquels l'hypothèse naïve explique bien les données, les modèles en grille sont moins performants, surtout quand il n'y a pas assez d'individus pour construire une grille suffisamment informative. A contrario, quand des dépendances complexes entre les variables explicatives sont incompatibles avec l'hypothèse naïve, les modèles en grille permettent une amélioration significative des performances.

La comparaison des performances des deux familles de modèles est en soi instructive, en apportant des informations sur la nature de la corrélation entre les variables explicatives et la variable à expliquer.

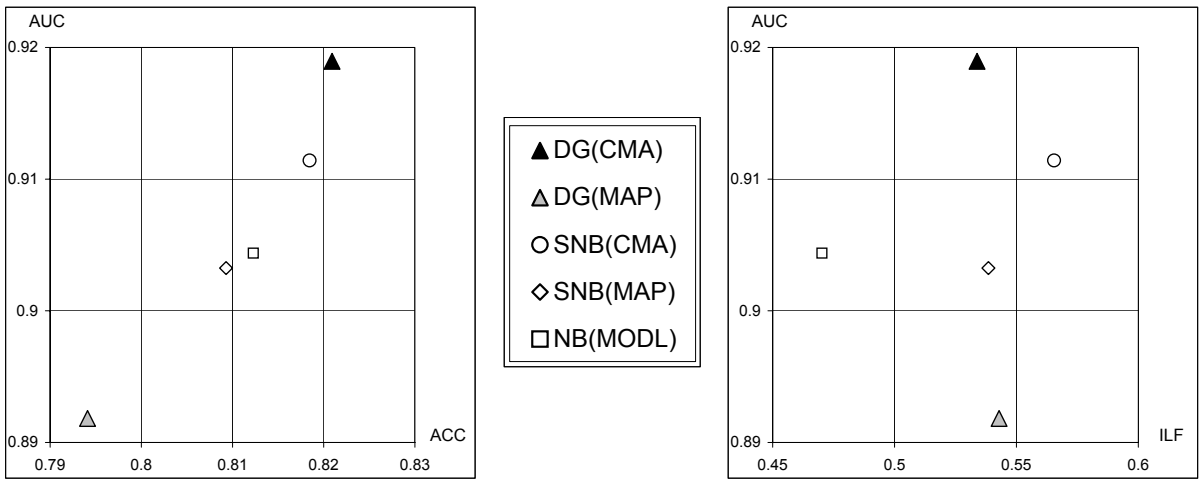


FIG. 4.10 – Moyenne des critères ACC, AUC et ILF pour l'évaluation de classifieurs en grille de données et Bayesiens naïfs sur 30 jeux de données de l'UCI.

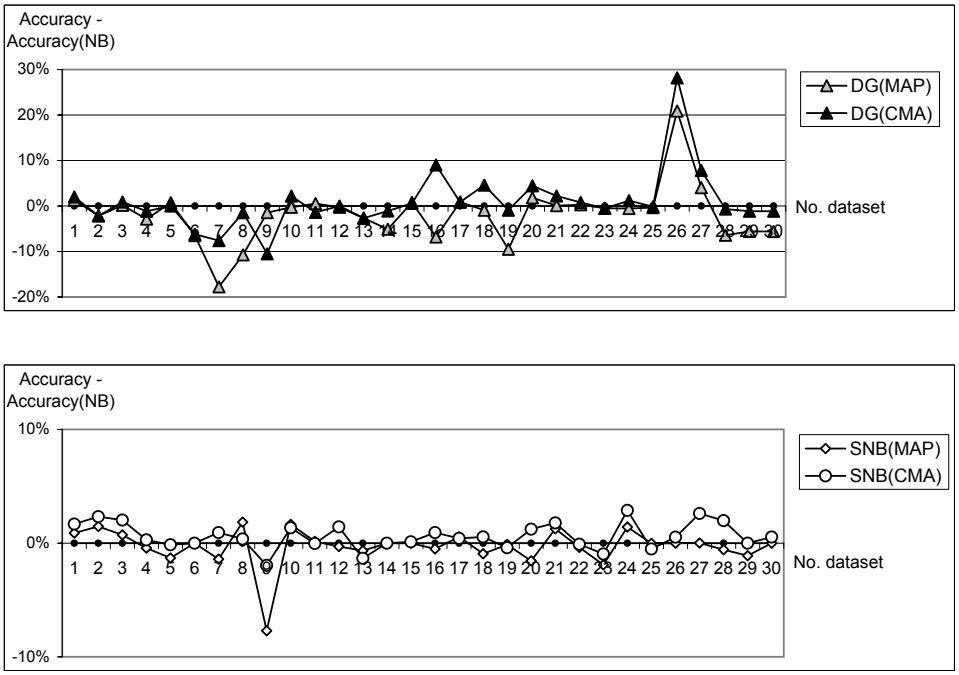


FIG. 4.11 – Apport en taux de prédiction par rapport au classifieur Bayesien naïf sur 30 jeux de données de l'UCI, pour les classifieurs en grille DG(CMA) et DG(MAP) et Bayesien naïf SNB(CMA) et SNB(MAP). Noter la différence d'échelle entre les deux figures.

Apport du moyennage de modèles en grille. Que ce soit pour les modèles Bayesiens naïfs ou les modèles en grille, le moyennage par taux de compression améliore fortement les performances prédictives par rapport au seul modèle MAP.

Dans le cas Bayesien naïf, l'apport est systématique, mais limité en raison de la faible expressivité de l'espace des modèles. Le moyennage étant effectué sur un grand nombre de modèles, les performances ne sont pas améliorées quand on accroît le temps d'apprentissage.

A l'inverse, dans le cas des modèles en grille, l'apport est variable selon les jeux de données, avec des différences de forte amplitude. Cela est dû à la forte expressivité de l'espace des modèles en grille. L'apport du moyennage est ici limité par le faible nombre de modèles moyennés (moins d'une centaine en moyenne). Des expérimentations, non reproduites ici, montrent que le moyennage sur un plus grand nombre de modèles permet d'améliorer les performances, au prix d'un surcoût à la fois en temps d'apprentissage et en temps de déploiement des modèles.

4.4.1.5 Evaluation de l'apport explicatif

On s'intéresse ici à l'apport explicatif des modèles, en privilégiant leur interprétabilité plutôt que leur performance prédictive. On choisit de comparer à cet effet le classifieur en grille DG(MAP) et le classifieur Bayesien naïf SNB(MAP). Tous les deux utilisent une approche Bayesienne de la sélection de modèles et aboutissent à une sélection de variables facilement interprétable.

Le classifieur en grille recherche une estimation de densité conditionnelle s'exprimant conjointement sur l'ensemble des variables sélectionnées, et sélectionne le meilleur modèle en tenant compte de l'incertitude sur tous les paramètres des modèles en grille.

Le classifieur Bayesien naïf sélectif recherche une estimation de densité conditionnelle se factorisant sur les variables sélectionnées, et sélectionne le meilleur modèle en tenant compte de l'incertitude sur la sélection de variables, mais en ignorant l'incertitude sur les estimateurs de densité conditionnelle univariée.

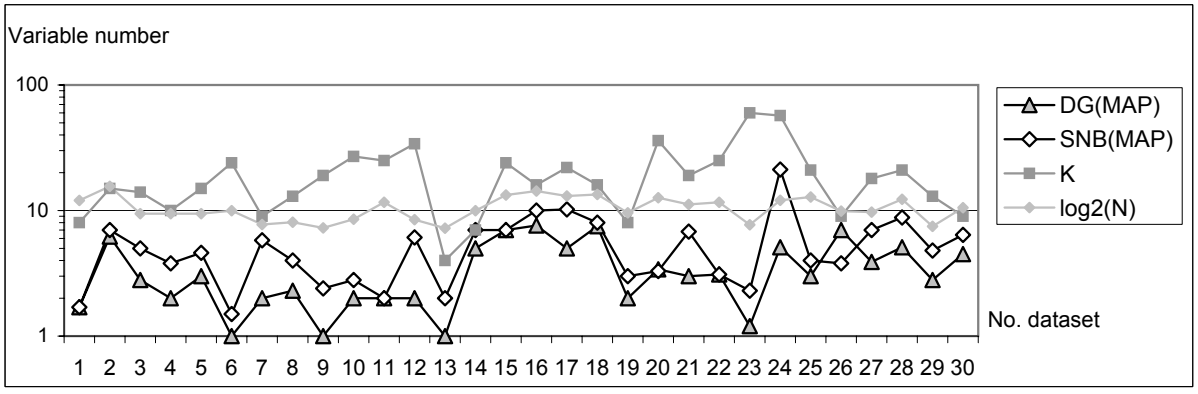


FIG. 4.12 – Moyenne du nombre de variables sélectionnées sur 30 jeux de données de l'UCI, pour les classifieurs en grille DG(MAP) et Bayesien naïf SNB(MAP). A titre de référence, le nombre de variables initiales K ainsi que le logarithme du nombre d'individus $\log_2 N$ sont également tracés.

La figure 4.12 montre les nombres de variables sélectionnées pour chacun des classi-

fieurs par grille et Bayesien naïf. Les modèles en grille DG(MAP) sélectionnent seulement 3.5 variables en moyenne, nettement moins que les 5.5 variables en moyenne des modèles Bayesien naïfs SNB(MAP).

En dépit de cette différence entre nombres de variables sélectionnées, on ne peut privilégier l'un des modèles au détriment de l'autre. Chaque modèle propose une interprétation différente, dont la validité peut s'évaluer par les performances prédictives ou mieux par la vraisemblance. Un modèle en grille s'interprète au moyen de l'ensemble de ses variables conjointement, par l'intermédiaire des cellules qui mettent en évidence les corrélations. Un modèle Bayesien naïf s'interprète au moyen de l'ensemble de ses variables considérées individuellement et indépendamment. Cette interprétation, plus simple, conduit à accepter un plus grand nombre de variables explicatives.

Grille de données pour la base Mushroom. La base Mushroom comporte 8416 individus, 22 variables explicatives catégorielles et une variable à expliquer booléenne. La meilleure grille obtenue sur l'ensemble de tous les individus est basée sur le groupement de valeurs de cinq variables explicatives. On décrit ci-dessous les groupes de valeurs pour chacune de ces variables, en désignant chaque groupe par le nom de sa valeur la plus fréquente.

- Odor :
  . None = {None, Anise, Almond, Musty},
  . Foul = {Foul, Fishy, Spicy, Pungent, Creosote},
- SporePrintColor :
  . Brown = {Brown, Black, Chocolate, Green, Buff, Purple, Orange, Yellow},
  . White = {White},
- Population :
  . Several = {Several, Solitary, Scattered, Abundant, Numerous},
  . Clustered = {Clustered},
- StalkRoot :
  . Bulbous = {Bulbous, Equal, Club, Rooted},
  . ? = {?},
- RingNumber :
  . One = {One, None},
  . Two = {Two}.

On présente dans le tableau 4.2 le modèle en grille correspondant, qui ne comporte que 12 cellules non vides alors que la grille complète en comprend $32 = 2^5$. Les fortes corrélations détectées permettent de séparer parfaitement les valeurs à expliquer. La variable Odor est fortement informative à elle seule, puisque tous les champignons du groupe "Foul" de la variable Odor sont toxiques. Les autres champignons, du groupe majoritaire "None", sont presque tous comestibles, exceptés ceux de trois cellules représentant moins de 2% des champignons. Ces trois cellules nécessitent une interaction complexe entre les quatres variables explicatives restantes pour être catactérisées.

TAB. 4.2 – Classification des champignons de la base Mushroom en edible ou poisonous par cellule d'une grille basée sur cinq variables explicatives.

Odor	Spore Print Color	Population	Stalk Root	Ring Number	Class		Effectif
					edible	poisonous	
None	Brown	Several	Bulbous	One	100%	0%	3648
Foul	Brown	Several	Bulbous	One	0%	100%	2032
Foul	White	Several	?	One	0%	100%	1728
None	White	Several	?	Two	100%	0%	288
None	White	Clustered	?	Two	100%	0%	192
None	Brown	Several	?	One	100%	0%	144
None	Brown	Clustered	?	One	100%	0%	96
None	Brown	Several	Bulbous	Two	0%	100%	72
None	White	Several	Bulbous	One	100%	0%	72
None	White	Clustered	Bulbous	One	0%	100%	64
None	White	Several	Bulbous	Two	100%	0%	48
None	White	Several	?	One	0%	100%	32

4.4.2 Classification non supervisée

On présente dans cette section des résultats préliminaires sur l'utilisation des modèles en grille pour la classification non supervisée. Des exemples d'application de la méthode sont présentés sur quelques jeux de données à titre illustratif.

4.4.2.1 Classification non supervisée par grille

Le critère d'évaluation utilisé est celui des modèles en grille pour la classification non supervisée, définis en section 2.4 (formule multivariée avec sélection de variables 2.37). La méthode d'optimisation, présentée en section 3.2, est la même que dans le cas supervisé. On retient ici le meilleur modèle issu de l'optimisation pour construire une partition des individus en un ensemble de cellules disjointes.

Le modèle en grille s'interprète comme un estimateur non paramétrique de densité jointe, permettant de détecter les corrélations entre les variables. Ces corrélations sont capturées dans les cellules de la grille, qui regroupent des individus distribués de façon similaire sur chaque variable de la grille. On retrouve ainsi l'objectif classique des méthodes de classification non supervisée, qui visent à regrouper les individus similaires en régions homogènes, les régions étant les plus différentes possible.

4.4.2.2 Apport explicatif des modèles en grille

On utilise à nouveau les 30 jeux de données de l'UCI décrits dans le tableau 3 de l'annexe C, et on lance pour chacun d'eux l'apprentissage d'un modèle en grille non supervisé sur l'ensemble de tous les individus. La figure 4.13 présente pour chaque jeu de données le nombre de variables sélectionnées ainsi que le nombre de cellules non vides.

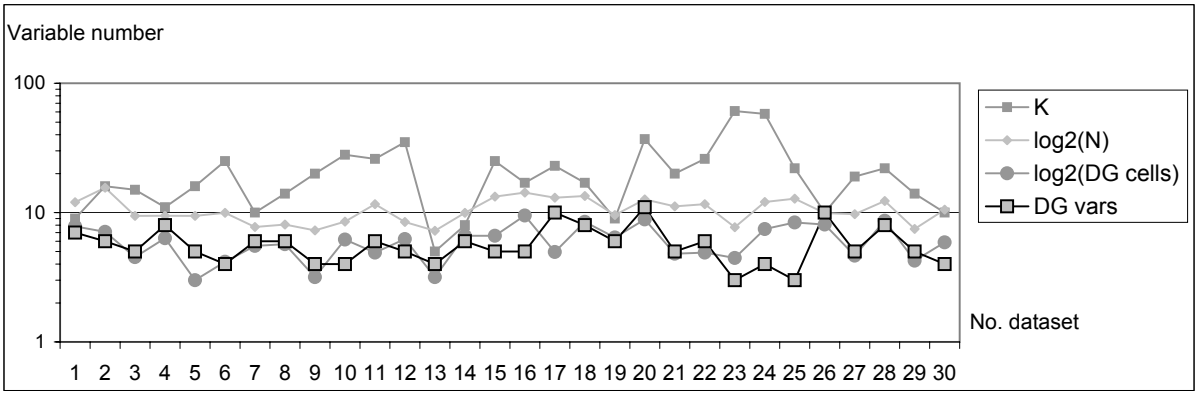


FIG. 4.13 – Moyenne du nombre de variables et de cellules non vides des grilles non supervisées sur 30 jeux de données de l'UCI. A titre de référence, le nombre de variables initiales K ainsi que le logarithme du nombre d'individus $\log_2 N$ sont également tracés.

Par rapport au cas supervisé, il s'agit de détecter le maximum de corrélations entre toutes les variables, et non uniquement les corrélations entre les variables explicatives et une seule variable à expliquer. Les grilles obtenues dans le cas non supervisé détectent plus d'informations, ce qui se traduit par un nombre moyen de variables sélectionnées égal à 6, au lieu de 3.5 seulement dans le cas supervisé.

De façon conforme aux seuils d'apprentissage de la section 4.1, le nombre de variables sélectionnées est toujours inférieur au log base 2 du nombre d'individus, et le nombre de cellules au nombre d'individus.

Interprétation sous forme d'un ensemble de règles. La partition des individus en cellules étant factorisable sur les partitions univariées des variables sélectionnées, le modèle est aisément interprétable. En effet, chaque partition univarié s'interprète comme une disjonction de régles élémentaires, du type $v_1 \leq x < v_2$ pour les intervalles numériques ou $x \in \{v_1, v_2, \ldots\}$ pour les groupes de valeurs catégorielles. Une cellule, résultant du produit cartésien des partitions univariées, s'interprète comme une conjonction de règles élémentaires univariées. La grille globale correspond alors à un ensemble de règles, s'exprimant toutes sur la même base de règles élémentaires univariées. Des exemples illustratifs sont présentés en section 4.4.2.3.

4.4.2.3 Quelques exemples de grilles non supervisées

A titre illustratif, on présente des exemples des modèles en grille pour l'analyse non supervisée des jeux de données Iris, Mushroom et Adult.

Base Iris. La base Iris comporte 150 individus pour cinq variables, quatre numériques et une catégorielle. La grille obtenue partitionne l'espace des individus en 9 cellules non vides, basée sur 4 variables. Les partitions univariées de ces variables sont :

– Class : 3 groupes mono-valeur {Iris-setosa}, {Iris-versicolor} et {Iris-virginica},

– PetalLength : 3 intervalles $]-\infty; 2.45]$, $]2.45; 4.75]$ et $]4.75; +\infty[$,
– PetalWidth : 3 intervalles $]-\infty; 0.8]$, $]0.8; 1.75]$ et $]1.75; +\infty[$,
– SepalLength : 2 intervalles $]-\infty; 5.55]$ et $]5.55; +\infty[$.

Il est à noter que la grille contient $54 = 3 * 3 * 3 * 2$ cellules dont seulement 9 sont non vides. Ces cellules, reproduites dans le tableau 4.3 mettent en évidence de fortes corrélations entre les variables.

TAB. 4.3 – Grille non supervisée pour la base Iris.

Class	PetalLength	PetalWidth	SepalLength	Effectif
{Iris-setosa}	$]-\infty; 2.45]$	$]-\infty; 0.8]$	$]-\infty; 5.55]$	47
{Iris-virginica}	$]4.75; +\infty[$	$]1.75; +\infty[$	$]5.55; +\infty[$	45
{Iris-versicolor}	$]2.45; 4.75]$	$]0.8; 1.75]$	$]5.55; +\infty[$	33
{Iris-versicolor}	$]2.45; 4.75]$	$]0.8; 1.75]$	$]-\infty; 5.55]$	11
{Iris-versicolor}	$]4.75; +\infty[$	$]0.8; 1.75]$	$]5.55; +\infty[$	5
{Iris-virginica}	$]4.75; +\infty[$	$]0.8; 1.75]$	$]5.55; +\infty[$	4
{Iris-setosa}	$]-\infty; 2.45]$	$]-\infty; 0.8]$	$]5.55; +\infty[$	3
{Iris-versicolor}	$]4.75; +\infty[$	$]1.75; +\infty[$	$]5.55; +\infty[$	1
{Iris-virginica}	$]2.45; 4.75]$	$]0.8; 1.75]$	$]-\infty; 5.55]$	1

Chaque cellule de la grille peut être caractérisée par une règle simple. Par exemple, les individus des deux cellules de plus fort effectif du tableau 4.3 sont décrits précisément par la règle 1 sur les petites fleurs et la règle 2 sur les grandes fleurs.

$$\begin{aligned} \text{Règle 1} \;:\; & \text{Class} \in \{Iris-setosa\} \\ & \text{PetalLength} \in \,]-\infty; 2.45] \\ & \text{PetalWidth} \in \,]-\infty; 0.8] \\ & \text{SepalLength} \in \,]-\infty; 5.55] \end{aligned}$$

$$\begin{aligned} \text{Règle 2} \;:\; & \text{Class} \in \{Iris-virginica\} \\ & \text{PetalLength} \in \,]4.75; +\infty[\\ & \text{PetalWidth} \in \,]1.75; +\infty[\\ & \text{SepalLength} \in \,]5.55; +\infty[\end{aligned}$$

La grille complète est ainsi décrite par un ensemble de règles intelligibles, puisque s'exprimant toutes sur les mêmes variables et les mêmes parties univariées.

Base Mushroom. La base Mushroom comporte 8416 individus pour 23 variables, toutes catégorielles. La grille obtenue capture des corrélations entre 10 des 23 variables, ce qui correspond à un nombre important de variables puisque l'on n'est pas loin de $13 \approx \log_2 8416$. Parmi les 10 variables de la grille, 6 comportent deux groupes de valeurs et 4 trois groupes de valeurs, ce qui produit une grille de $5184 = 2^6 3^4$ cellules. Cette grille

ne contient que 31 cellules non vides, ce qui traduit de très fortes corrélations entre les variables.

A titre anecdotique, il est intéressant de noter une certaine régularité dans les effectifs des 31 cellules de la grille : 1728, 1728, 1024, 864, 560, 432, 288, 192, 144, 144, 128, 128, 96, 96, 96, 96, 72, 72, 72, 72, 48, 48, 48, 48, 48, 48, 32, 24, 24, 8, 8. Cette régularité semble surprenante, ce qui laisse planer un doute sur le qualificatif "réelle" pour la base Mushroom.

Pour procéder à une analyse descriptive plus complète et détecter des corrélations entre toutes les variables, une seule grille ne peut suffire puisque le nombre de variables excède le log base 2 du nombre d'individus. Il faudrait alors recourir à des grilles multiples, en recouvrement ou non. On se rapproche alors des méthodes de recherche de règles d'association initiées par Agrawal [Agrawal *et al.*, 1993].

Base Adult. La base Adult comporte 48842 individus pour 16 variables, pour moitié numériques et catégorielles. La grille obtenue n'utilise que 6 variables pour environ 3500 cellules, dont seulement 137 sont non vides. Les dix cellules de plus fort effectif, présentées dans le tableau 4.4, représentent environ 60% des individus.

Comme dans le cas précédent, des grilles multiples permettraient de capturer d'avantage de corrélations. Par exemple, on peut rechercher les grilles bivariées de façon exhaustives pour toutes les paires de variables pour détecter toutes les corrélations élémentaires, voire les variables redondantes. Cette méthode permet par exemple de détecter automatiquement que les variables numérique educationNum et catégorielle education sont redondantes.

On peut aussi post-traiter directement la grille multivariée du tableau 4.4 pour obtenir une grille réduite, ce qui en facilite l'interprétation. De telles grilles réduites peuvent être obtenues en imposant une taille maximale aux partitions univariées ou en projetant la grille sur un nombre réduit de variables.

TAB. 4.4 – Grille non supervisée pour la base Adult.

class	education	edNum	maritalStatus	relationship	sex	Effectif
{less}	{HS-grad}	9	{Married...}	{Husband}	{Male}	4382
{less}	{HS-grad}	9	{Never-married...}	{Unmaried...}	{Female}	4205
{less}	{HS-grad}	9	{Never-married...}	{Unmaried...}	{Male}	4071
{less}	{Some-college}	10	{Never-married...}	{Unmaried...}	{Female}	3595
{less}	{Some-college}	10	{Never-married...}	{Unmaried...}	{Male}	2858
{more}	{Bachelors}	13	{Married...}	{Husband}	{Male}	2462
{less}	{Some-college}	10	{Married...}	{Husband}	{Male}	2050
{more}	{HS-grad}	9	{Married...}	{Husband}	{Male}	2005
{less}	{Bachelors}	13	{Never-married...}	{Unmaried...}	{Female}	1793
{more}	{Some-college}	10	{Married...}	{Husband}	{Male}	1608

4.4.3 Coclustering des individus et variables

On présente dans cette section une application particulière des modèles en grille de classification non supervisée pour le coclustering des individus et des variables d'un jeu de données. Après avoir introduit le coclustering et la méthode par grille correspondante, on en présente une utilisation dans le cadre de l'apprentissage semi-supervisé. Les méthodes présentées sont illustrées au moyen d'un problème de classification de textes.

4.4.3.1 Coclustering par grille

Un coclustering [Hartigan, 1972] est défini comme le regroupement simultané des lignes et des colonnes d'une matrice. Dans le cas des jeux de données de faible densité, ayant de nombreux 0 dans le tableau croisé individus * variables, le coclustering est une technique attractive pour identifier des corrélations entre groupes d'individus et groupes de variables.

Considérons un jeu de données comportant N individus $\mathrm{Id}1, \mathrm{Id}2, \ldots, \mathrm{Id}N$ et K variables $\mathrm{Var}1, \mathrm{Var}2, \ldots, \mathrm{Var}K$. Le tableau croisé individus * variables comporte NK valeurs v_{nk}. Soit V le nombre de valeurs non nulles de ce tableau croisé. Quand $V \ll NK$, le tableau croisé correspond à une matrice creuse, et on peut représenter l'emplacement des valeurs non vides sous la forme d'un nouveau tableau ne comportant que trois colonnes, "Individu", "Variable" et "Valeur", et V lignes. Chaque ligne de ce tableau contient un triplet du type $(\mathrm{Id}n, \mathrm{Var}k, v_{nk})$, correspondant à une valeur v_{nk} non vide. Dans le cas de valeurs v_{nk} booléenne, ou si l'on remplace les valeurs v_{nk} par un indicateur booléen absence/présence, on peut réduire le tableau de triplets $(\mathrm{Id}n, \mathrm{Var}k, v_{nk})$ à un tableau de paires $(\mathrm{Id}n, \mathrm{Var}k)$. Ce tableau à deux colonnes est plein, de taille $V * 2$, par opposition au tableau creux initial, de taille $N * K$, comme illustré sur la figure 4.14.

	Var_1	Var_2	Var_3	Var_4	Var_5	...
Id_1	0	1	0	0	0	...
Id_2	0	0	1	1	0	...
Id_3	0	1	0	0	0	...
Id_4	0	0	0	1	1	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱

⟶

ID Individu	*ID Variable*
Id_1	Var_2
Id_2	Var_3
Id_2	Var_4
Id_3	Var_2
Id_4	Var_4
Id_4	Var_5
⋮	⋮

FIG. 4.14 – Jeu de données binaire de faible densité : depuis la matrice creuse (individus * variables) au tableau bivarié dense.

En appliquant l'analyse non supervisée par grille à ce tableau plein, on obtient une grille bivariée basée sur le groupement des individus d'une part, le groupement des variables d'autre part. Les critères d'évaluation des grilles conduisent à maximiser la corrélation entre groupes d'individus et groupes de variables, ce qui correspond à l'objectif général du coclustering. Il est à noter que les coclusters sont ici les cellules de la grille, et qu'ils forment une partition sans recouvrement du jeu de données.

4.4.3.2 Application du coclustering à l'apprentissage semi-supervisé

Le coclustering permet d'appliquer une approche semi-supervisé [Chapelle *et al.*, 2006] de façon à exploiter toutes les données disponibles, qu'elles soient étiquetées ou non (étiquette signifie ici valeur à expliquer). A cet effet, on constitue un jeu de données contenant toutes les données disponibles en apprentissage et test, en supprimant la variable à expliquer. En appliquant la méthode de coclustering précédemment décrite, on recherche les motif "naturels" du jeu de données.

Une fois le prétraitement non supervisé de coclustering effectué, l'étape d'apprentissage supervisé consiste à étiqueter chaque groupe d'individus identifié lors du coclustering à l'aide des individus étiquetées disponibles en apprentissage. L'étape de prédiction se fait en recherchant pour chaque individu en test son groupe d'individu étiqueté le plus proche (au sens du critère d'évaluation de la grille de coclustering), et à lui associer la distribution des étiquettes de ce groupe d'individus.

Le prétraitement des données de façon semi-supervisée par coclustering n'a de sens que sous l'hypothèse que les motifs "naturels" du jeu de données soient corrélés avec les valeurs à expliquer (motifs "prédéfinis"). Cette hypothèse paraît raisonnable pour certains jeux de données, notamment dans le domaine de la reconnaissance des formes.

4.4.3.3 Exemple illustratif

La base Reuters [Reuters, 2000] est une base d'articles en langue anglaise, dont nous avons extrait un petit corpus de 4500 textes classés en cinq catégories selon les rubriques Reuters : Sport, Emploi, Art, Catastrophes et Santé. La représentation utilisée est la représentation "sac de mots", après passage en minuscules, ce qui correspond à un total d'environ 41000 mots utilisés dans notre corpus. Chaque texte est ainsi représenté comme un vecteur de taille 41000, dont chaque valeur booléenne vaut 1 si un mot est utilisé dans le texte. Le tableau croisé initial contient ainsi 4500 individus, les textes, et 41000 variables, les mots. Ce tableau croisé est creux avec un taux de remplissage de 0.4%, pour environ 700000 valeurs non vides.

Les valeurs non vides sont représentables sous la forme d'un tableau à deux colonnes Texte * Mot, comportant 700000 lignes. En appliquant la technique de coclustering par grille précédemment décrite, on obtient environ 150 groupes de textes et 350 groupes de mots. La grille bivariée résultante n'est plus creuse, puisqu'elle contient 50000 cellules dont 40000 non vides, avec en moyenne une vingtaine de paires (Texte, Mot) par cellule.

La granularité du coclustering est très fine. Chaque groupe de mots correspond à des mots statistiquement "proches", dans le sens où ils sont distribués de la même façon sur les groupes de textes. On trouve des groupes de mots à faible sémantique, dont la distribution est proche de l'uniforme sur l'ensemble des groupes de textes. C'est le cas par exemple des groupes de mots :

- {the, of, in, on, for, ...},
- {about, some, other, most, where, ...},
- {year, last, week, since, most, ...}.

A l'inverse, certains groupes de mots, ayant des distributions fortement piquées, semblent capturer des informations de nature sémantique proches des rubriques Reuters prédéfinies.

C'est le cas par exemple des groupes de mots :

- {crash, crashed, plane, flight, airport, ...}.
- {cup, match, play, game, club, ...}.
- {drug, patients, treatment, blood, cancer, ...}.

Chaque groupe de textes correspond à des textes "proches" dans le sens où ils sont distribués de façon similaire sur les groupes de mots. Afin de vérifier que les groupes de textes "naturels" identifiés par le coclustering sont corrélés avec les rubriques prédéfinies de la base Reuters, on applique l'approche semi-supervisée décrite précédemment. La base est découpée en 70% pour l'apprentissage et 30% pour le test. Le coclustering est effectué sur toute la base, apprentissage et test, en ignorant les rubriques Reuters. Pour chaque texte à classer en test, on recherche le groupe de textes en apprentissage le plus proche, et on lui associe la rubrique Reuters majoritaire de ce groupe. Le taux de bonne prédiction obtenu est de 91.5%, à comparer avec celui obtenu par un classifieur Bayesien naïf NB(MODL) de 93%.

Les performances prédictives issues du coclustering sont ici compétitives, ce qui signifie qu'il y a une forte corrélation entre les groupes de textes "naturels" et les rubriques Reuters prédéfinies. Ce qui est également intéressant est que les groupes de mots identifiés semblent pertinents pour extraire automatiquement une partie de la sémantique d'une base de textes.

Il est à noter que l'optimisation des grilles dans le cas du coclustering des textes et mots de la base Reuters est un problème difficile, puisque l'on recherche une solution dans un espace dont la taille avoisine 10^{150000}. Tous les algorithmes d'optimisation gloutonne, de pré-optimisation et post-optimisation et de méta-heuristique décrits en annexe D sont nécessaires pour obtenir une grille bivariée meilleure que la grille nulle. L'apprentissage nécessite quelques heures de calcul intensif sur une machine moderne (PC 3.2 Ghz), jusqu'à plusieurs jours selon le niveau d'optimisation souhaité.

4.4.4 Bilan de l'évaluation des modèles en grille

Dans cette section, nous avons proposé une mise en oeuvre des modèles en grille dans le cadre de la classification supervisée, non supervisée et du coclustering, ce qui correspond aux sections 2.1 et 2.4 du chapitre 2. Bien que non exhaustive, cette évaluation permet de dégager certaines caractéristiques générales des modèles en grille.

Les grilles obtenues sont fortement informatives, et permettent en sélectionnant un très faible nombre de variables de construire des méthodes prédictives compétitives, comme le montrent les évaluations comparatives avec le classifieur Bayesien naïf.

Les modèles en grille permettent d'identifier automatiquement des groupes de variables fortement corrélées, ce qui présente un intérêt en soi pour la préparation des données. Une grille peut ainsi être utilisée pour la construction automatique de nouvelles variables, plus informatives que les variables de la représentation initiale.

Les modèles en grille ont un fort intérêt sur le plan explicatif, notamment dans le cadre de l'analyse exploratoire avec les grilles non supervisées et les grilles de coclustering. Les modèles proposés sont facilement interprétables sous la forme d'ensembles de règles s'exprimant toutes sur les mêmes variables.

Les expérimentations préliminaires sur le moyennage de grille ont validé le potentiel d'amélioration des performances prédictives. En dépit du côté rudimentaire de la méthode proposée, l'apport du moyennage est incontestable

Globalement, ces premières évaluations sont très positives et confirment le potentiel théorique des modèles en grille. Néanmoins, d'importants travaux restent nécessaires pour confirmer ces résultats d'évaluation et exploiter au mieux les bénéfices potentiels des modèles en grille en développant de nouvelles méthodes de préparation, modélisation et interprétation des données.

4.5 Évaluation sur des challenges internationaux

Cette section résume l'ensemble des résultats de participation à des challenges internationaux, visant à évaluer les méthodes de classification basées sur les modèles en grille de façon comparative avec les méthodes de l'état de l'art.

4.5.1 Intérêt des challenges internationaux

L'objectif des challenges internationaux est de stimuler la recherche et de permettre une évaluation comparative des méthodes de l'état de l'art, en les confrontant à des problèmes difficiles. Parmi les nombreux facteurs influant sur la qualité d'un challenge, citons :

- le nombre et la difficulté des problèmes à résoudre, qui doivent être représentatifs et sélectifs, sans toutefois être dissuasifs pour les participants,
- les critères d'évaluation, certains faciles à mesurer (performance prédictive, robustesse...), d'autres moins (facilité de mise en oeuvre d'une méthodes, temps d'apprentissage, interprétabilité des résultats...),
- le nombre et la diversité des méthodes alternatives évaluées, en relation directe avec le nombre de participants,
- l'implication des participants, nécessaire pour la mise en oeuvre efficace des méthodes complexes, mais donnant lieu au dilemme de l'évaluation de la qualité d'une méthode (préférable) ou de l'implication du participant (à éviter).

Ces facteurs entrent en concurrence et l'organisation de challenges pertinents relève de l'art du compromis. Un critère incontestable permet de trancher la question de l'intérêt d'un challenge : un challenge est utile s'il permet d'apprendre et de progresser. A ce titre, les challenges présentés ci-dessous ont tous été instructifs.

4.5.2 Feature Selection Challenge

Le Feature Selection Challenge [Guyon, 2003], organisé au sein de la conférence NIPS 2003, a pour but d'évaluer l'intérêt de la sélection de variables pour la classification supervisée.

Les jeux de données, résumés dans le tableau 4.5, comportent un très grand nombre de variables, dont une partie (% Probe), indépendante de la variable à expliquer, a été ajoutée pour augmenter la difficulté des problèmes.

Les critères d'évaluation utilisés sont :
- BER : Balanced Error Rate, utile pour évaluer le taux d'erreur dans le cas de distributions déséquilibrées,
- AUC : Area Under the ROC curve,
- Ffeat : fraction des variables sélectionnées,
- Fprobe : fraction des variables inutiles sélectionnées.

Trois des critères, BER, Ffeat et Fprobe, organisés de façon hiérarchique, sont utilisés pour classer les participants. Globalement, 135 groupes de recherche ont participé au challenge, mais seulement une vingtaine ont soumis des solutions complètes pour le classement final. Les détails du challenge sont disponibles sur le site `http ://www.nipsfsc.ecs.soton.ac.uk/`, et commentés dans le livre [Guyon *et al.*, 2006a].

Tab. 4.5 – Jeux de données du Feature Selection Challenge.

Dataset	Variables		Individus		
	Total	% Probe	Train	Valid	Test
Arcene	10000	30%	100	100	700
Gisette	5000	50%	6000	1000	6500
Dexter	20000	50%	300	300	2000
Dorothea	100000	50%	800	350	800
Madelon	500	96%	2000	600	1800

Notre soumission, baptisée ESNB, est basée sur la méthode de discrétisation univariée MODL (cf. section 4.2.1), utilisée comme prétraitement pour le classifieur Bayesien naïf avec sélection de variables basée sur l'optimisation du critère AUC (cf. section 4.3.2.3). La méthode est utilisée telle quelle, ou en prétraitement de sélection de variables pour un réseau de neurones (soumission ESNB+NN). Ces soumissions sont décrites dans [Boullé, 2006a].

Notre soumission en tant que groupe termine 14^{e} en moyenne selon le classement final du challenge. La meilleure performance est obtenue sur le jeu de données Dorothea comportant 100000 variables, où la méthode termine 4^{e}.

Bien que le réseau de neurones améliore significativement les performances prédictives (BER) par rapport à celles du classifieur Bayesien naïf, celles-ci restent environ deux fois moindres que celles du vainqueur du challenge.

En revanche, comme le montre la figure 4.15, la méthode sélectionne seulement 1% des variables en moyenne, soit de 10 à 100 fois moins que les méthodes en tête du challenge. De plus, une seule variable est sélectionnée à tort parmi les milliers de variables inutiles des jeux de données. Sur ce dernier critère (Fprobe), la méthode devance très largement toutes les méthodes alternatives, ce qui confirme la capacité de la méthode de discrétisation MODL à identifier les variables non informatives.

La participation à ce challenge a permis de valider la méthode de discrétisation MODL, et a mis en évidence les limites du classifieur Bayesien naïf sélectif avec optimisation du critère AUC. Les axes d'amélioration retenus portent principalement sur :

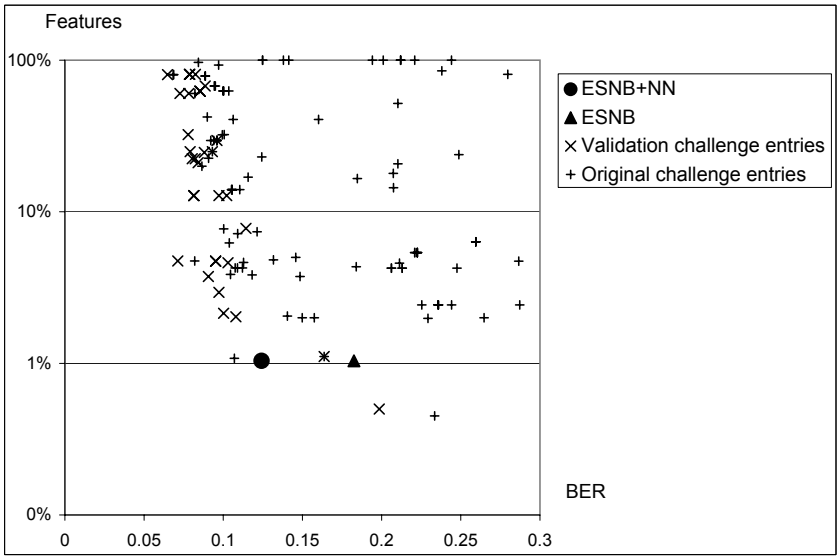


FIG. 4.15 – Analyse bi-critère de l'ensemble des soumissions au challenge, avec le critère BER en abscisse et le nombre de variables sélectionnées en ordonnée.

- l'efficacité en temps d'apprentissage,
- la robustesse de la sélection de variables,
- l'opportunité du moyennage de modèles,
- les limites de l'hypothèse naïve du classifieur utilisé.

4.5.3 Performance Prediction Challenge

Le Performance Prediction Challenge [Guyon *et al.*, 2006b], organisé au sein de la conférence IJCNN 2006, a pour but d'évaluer la capacité des méthodes de classification supervisée d'une part à obtenir de bonnes performances prédictives, d'autre part à prédire correctement leur performance future sur des données non vues en apprentissage.

Les jeux de données, résumés dans le tableau 4.6 sont caractérisés par un nombre d'individus en validation 10 fois inférieur à celui en apprentissage, ce qui est insuffisant pour effectuer une sélection de modèles, et un nombre d'individus en test 10 fois supérieure à celui en apprentissage, ce qui permet une évaluation fiable de la prédiction de la performance en test.

Les critères d'évaluation utilisés sont :

- BER,
- AUC,
- Guess error : différence entre le BER prédit et le BER obtenu en test.

Une combinaison de deux des critères, BER et Guess error, est utilisée pour classer les participants. Globalement, 145 groupes de recherche ont participé au challenge, et 28 d'entre eux ont soumis des solutions complètes pour le classement final. Les détails du challenge sont disponibles sur le site `http ://www.modelselect.inf.ethz.ch/index.php`.

Nos soumissions, préfixées par SNB(CMA), sont basées sur la méthode de discrétisa-

TAB. 4.6 – Jeux de données du Feature Selection Challenge.

Dataset	Variables	Individus		
		Train	Valid	Test
Ada	48	4147	415	41471
Gina	970	3153	315	31532
Hiva	1617	3845	384	38449
Nova	16969	1754	175	17537
Sylva	216	13086	1308	130858

tion univariée MODL (cf. section 4.2.1), utilisée comme prétraitement pour le classifieur Bayesien naïf avec sélection de variables basée sur une approche Bayesienne et moyennage de modèles par taux de compression (cf. sections 4.3.2.3 et 4.3.2.4).

Comme le critère d'évaluation principal est le BER, il n'est pas optimal de prédire la classe majoritaire à partir de l'estimation des probabilités conditionnelles. Dans le cadre du challenge, le seuil de prédiction est ajusté pour optimiser le BER, par post-processing des classifieurs, directement sur la base d'apprentissage.

La prédiction du BER futur est obtenue en utilisant une validation croisée stratifiée à 10 niveaux.

La méthode est utilisée telle quelle, ou en combinaison avec plusieurs scénarios de construction de variables. Ces ajouts de variables sont utilisés afin d'évaluer la tenue de charge des algorithmes en temps d'apprentissage, la robustesse de la sélection de variables et la capacité à contourner le biais du classifieur Bayesien naïf. Les scénarios de construction de variables, identiques pour tous les jeux de données, sont basés sur des sommes de paires ou triplets de variables tirées aléatoirement parmi les variables initiales. Trois scénarios de construction de variables sont utilisés, basés sur 10000 paires de variables ("10k F(2D)"), 100000 paires de variables ("100k F(2D)") et 10000 triplets de variables ("10k F(3D)"). Ces constructions de variables augmentent considérablement la taille de l'espace de représentation avec de nombreuses variables inutiles, redondantes ou informatives "par hasard", ce qui rend difficile la tâche de sélection de variables pour la classifieur Bayesien naïf utilisé. Ces soumissions sont détaillées dans [Boullé, 2006c, Boullé, 2007a].

Notre soumission en tant que groupe termine 7^{e} en moyenne selon le classement final du challenge, et 1re sur les deux jeux de données Ada et Sylva.

La figure 4.16 permet d'analyser les performances des méthodes sur les deux critères BER et Guess error conjointement. Cette figure montre que notre méthode est très robuste, avec de bonnes prédictions de sa performance en test. Sur ce critère seul, notre soumission est 4^{e} en moyenne. Sur le critère du BER, notre méthode reste à 2% des gagnants du challenge, ce qui la place en 11^{e} position parmi les participants. Il est à noter que sur le critère AUC, qui évalue l'ordonnancement des individus en test selon leur probabilité conditionnelle, notre méthode est classée 3^{e}.

L'analyse des scénarios de construction de variables permet de vérifier que les algo-

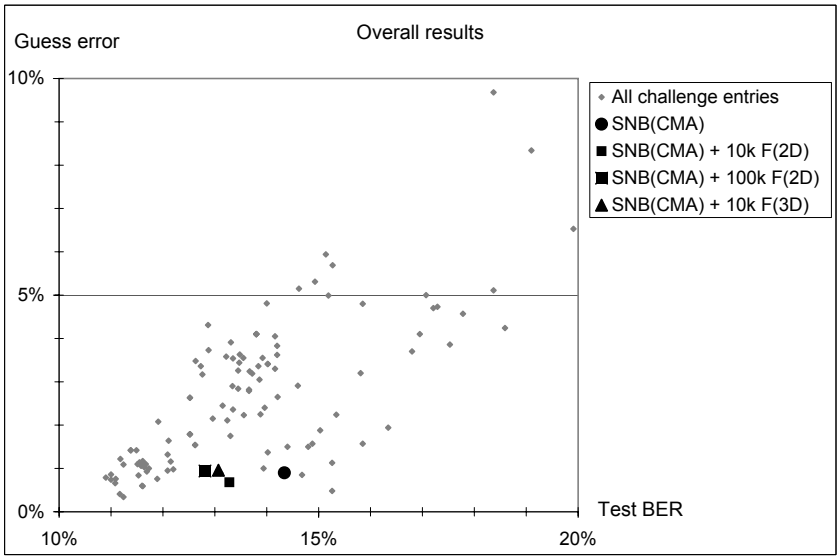


FIG. 4.16 – Analyse bi-critère de l'ensemble des soumissions au challenge, avec le critère BER en abscisse et le critère Guess error en ordonnée.

rithmes d'apprentissage ont une bonne tenue de charge, avec des temps de calcul quasi linéaires avec la taille des données. Les temps d'apprentissage mesurés sont conformes à leur complexité algorithmique théorique, en $O(NK \log(NK))$.

L'augmentation considérable de la taille de l'espace de représentation, d'un facteur 10 à 100, ne nuit jamais à la robustesse de la méthode SNB(CMA). Les performances prédictives ne sont jamais dégradées et elles sont parfois significativement améliorées si les nouvelles variables s'avèrent informatives. La construction de variables, même aléatoire, a ainsi donné lieu à une amélioration moyenne de 1.5% des performances prédictives, jusqu'à 5% pour le jeu de données Gina.

La participation à ce challenge a permis de valider les améliorations apportées à la sélection de variables pour le classifieur Bayesien naïf, principalement un critère de sélection de variables basé sur une approche Bayesienne, un algorithme d'optimisation plus efficace en temps de calcul, et une méthode de moyennage de modèles basée sur les taux de compression. Il semble que la méthode SNB(CMA) qui intègre ces améliorations ait atteint un niveau de maturité acceptable. Sa principale limite réside dans l'hypothèse naïve, qui empêche la méthode d'obtenir des performances compétitives s'il n'existe pas de sous-ensemble de variables compatible avec cette hypothèse dans l'espace de représentation. Deux pistes d'améliorations sont envisageables :

- augmenter la taille de l'espace de représentation par des techniques de construction de variables guidées par le domaine applicatif,
- réduire le biais de l'hypothèse naïve en recherchant des corrélations entre les variables.

4.5.4 Agnostic Learning vs. Prior Knowledge Challenge

L'Agnostic Learning vs. Prior Knowledge Challenge [Guyon *et al.*, 2007], organisé au sein de la conférence IJCNN 2007, a pour but d'évaluer l'impact de la connaissance du domaine applicatif sur les performances prédictives des méthodes de classification supervisée.

Les jeux de données, résumés dans le tableau 4.6, sont les mêmes que ceux du Performance Prediction Challenge. Il sont cette fois disponibles sous deux formats : *agnostic* et *prior*. Sous le format agnostic, toutes les variables explicatives sont numériques. Par rapport au Performance Prediction Challenge, les variables sont réordonnées aléatoirement, et les individus redistribués aléatoirement dans les ensembles d'apprentissage, de validation et de test. Sous leur format prior, la représentation est typée en variables numériques ou catégorielles (pour Ada, Gina et Sylva), ou disponible sous une forme native plus complexe (multi-tables pour Hiva et base de textes pour Nova).

Les critères d'évaluation sont le BER et l'AUC, mais seul le BER est utilisé pour le classement final des participants.

Le challenge se déroule en plusieurs étapes. Lors d'un premier jalon le 1er décembre 2006 (pour la conférence NIPS 2006), les participants sont évalués dans le cadre du challenge agnostic, afin de produire des performances de référence. Lors d'un second jalon le 1er mars 2007 (pour la conférence IJCNN 2007), les participants sont évalués dans le cadre du challenge prior. Les détails du challenge sont disponibles sur le site `http ://www.agnostic.inf.ethz.ch/`.

4.5.4.1 Jalon NIPS 2006

Pour le challenge NIPS 2006, nous avons utilisé exactement les mêmes méthodes que pour le Performance Prediction Challenge, à savoir le classifieur Bayesien naïf SNB(CMA) avec discrétisation MODL, sélection Bayesienne de variables et moyennage de modèles par taux de compression, accompagnés des mêmes scénarios de construction aléatoire de variables.

Les résultats de classement des participants sont disponibles sur `http ://clopinet.com/isabelle/Projects/NIPS2006/NIPSgame-Results-Dec-06.ppt`. De façon similaire au Performance Prediction Challenge, notre méthode arrive 6e en moyenne, 2e sur le jeu de données Ada et 1re sur Sylva.

4.5.4.2 Jalon IJCNN 2007

Pour ce second jalon, nous avons choisi de passer à l'évaluation des modèles en grille, dont les premières versions sont devenues opérationnelles in extremis quelques jours avant la date limite de soumission au challenge. Nous résumons ici notre participation au challenge, présentée dans [Boullé, 2007d].

Les méthodes de classification utilisées, décrites en section 4.4, sont :

- DG : classification par grille la plus probable,
- DGE : classification par moyennage de grille,
- DGCC : classification en mode semi-supervisé par coclustering des individus et des variables.

Les jeux de données Hiva et Nova ont un format complexe dans leur représentation prior : nous ne les avons pas utilisés dans ce format faute de temps et d'expérience dans les domaines applicatifs correspondants.

Les méthodes DG et DGE sont appliquées à tous les jeux de données restants. L'objectif est d'évaluer la tenue de charge des algorithmes et de confirmer l'apport du moyennage de modèles pour les jeux de données comportant de nombreuses variables. En effet, la méthode DG, limitée à des grille de $\log_2 N$ variables, ne peut ici rivaliser avec des méthodes exploitant un sous-ensemble de variables de plus grande taille.

Enfin, la méthode DGCC est appliquée sur les trois jeux de données dont le tableau individus * variables est creux : Gina dans son format prior, Hiva et Nova dans leur format agnostic.

Comme pour le précédent challenge, le BER est optimisé par post-processing des classifieurs. Le tableau 4.7 présente les résultats des méthodes pour le critère BER, évalués en validation croisée stratifiée à 10 niveaux sur les ensembles d'apprentissage et validation réunis. La colonne "Best" reproduit les meilleures performances obtenues sur ces jeux de données lors du précédent Performance Prediction Challenge.

TAB. 4.7 – Evaluation des performances prédictives des méthodes DG, DGE et DGCC basées sur les modèles en grille.

Name	Prior			Agnostic			
	DG	DGE	DGCC	DG	DGE	DGCC	Best
Ada	0.213	0.192		0.225	0.192		0.172
Gina	0.184	0.140	0.052	0.182	0.147		0.029
Hiva				0.340	0.310	0.320	0.276
Nova				0.243	0.135	0.075	0.044
Sylva	0.009	0.008		0.021	0.022		0.006

Les résultats de classement des participants sont disponibles sur `http ://clopinet.com/isabelle/Projects/agnostic/ALPK-Results-Mar-07.ppt`. Le challenge a attiré un total de 35 participants, avec 11 participants classés sur le challenge prior et 15 sur le challenge agnostic. Les organisateurs ont décidé de prolonger le challenge jusqu'au 1er août 2007, et seuls les classements des participants sont publiés avant cette date.

Les méthodes par grille obtiennent des résultat compétitifs, puisqu'elles sont classés 7e en moyenne sur le challenge agnostic et 3e sur le challenge prior. Les meilleurs classements par jeu de données (non attribuables actuellement à une méthode en grille particulière) sont 2e sur Ada et 3e sur Nova pour le challenge agnostic, 1er sur Ada et 4e sur Gina et Sylva pour le challenge prior.

Le challenge confirme les résultats obtenus avec les bases de l'UCI. Bien que les grilles seules soient parfois compétitives (sur Ada et Sylva, surtout dans leur format prior), le moyennage par grille améliore fortement les performances prédictives. La classification semi-supervisée par coclustering semble prometteuse au vu des résultats en validation croisée : pour Gina et Nova, les résultats sont nettement meilleurs que ceux obtenus précédemment par les méthodes basées sur les grilles ou sur le classifieur Bayesien naïf.

Il est néanmoins nécessaire d'attendre la publication définitive des résultats du challenge en test pour confirmer ces résultats préliminaires.

Analysons en détail les performances des modèles en grille sur la base Nova. Cette base qui contient environ 1700 textes pour 17000 variables (les mots) est la plus difficile pour les modèles en grille, limités à un sous-ensemble de variables de taille $10 \approx \log_2 1700$. La figure 4.17 présente les performances en validation croisée en fonction du temps d'apprentissage et du nombre total des variables du modèle moyenné DGE.

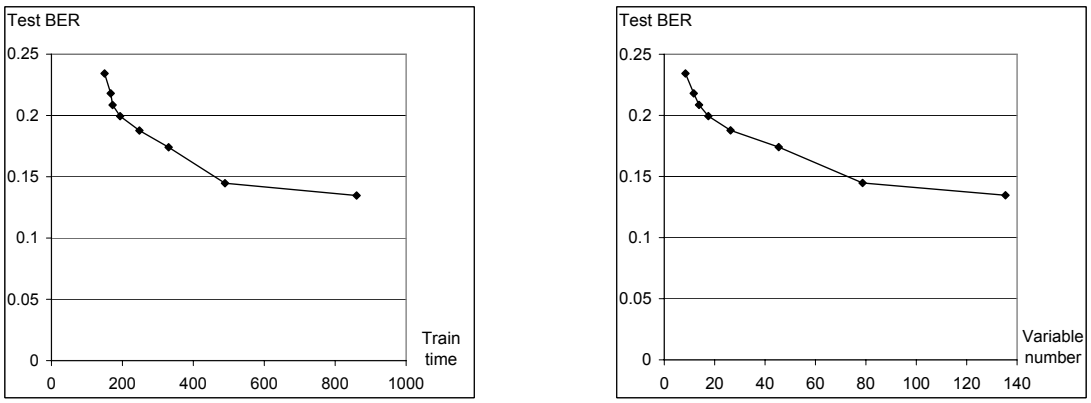


FIG. 4.17 – Performance prédictive en test en fonction du temps d'apprentissage à gauche et du nombre total de variables sélectionnées pour le classifieur DGE, sur la base Nova.

Il faut environ 100 secondes pour procéder à l'analyse univariée complète de la base et 40 secondes supplémentaires pour identifier le premier modèle en grille. Cette première grille comprend 8 variables et obtient un BER de 24%, soit deux fois mieux qu'une prédiction aléatoire. La meilleure grille obtenue au cours de toute l'optimisation n'améliore cette performance que de 1%.

Au bout de 850 secondes, quelques centaines de grilles ont été obtenues pour le moyennage de modèles, mais elles représentent au total seulement 140 variables, soit moins de 1% de l'ensemble des variables. Le modèle moyenné ne fait que s'améliorer avec le temps d'apprentissage et le nombre de grilles, et atteint un BER de 13% au moment où on a arrêté l'apprentissage. Il existe encore un potentiel d'amélioration important, mais les algorithmes d'optimisation des grilles, efficaces sur des jeux de données ayant de faibles nombres de variables, doivent être adaptés pour produire plus rapidement une plus grande diversité de grilles.

La méthode par coclustering nécessite plusieurs heures d'apprentissage pour modéliser en semi-supervisé les interactions entre les 17000 mots et les 19000 textes (bases d'apprentissage, validation et test réunies) de la base Nova. Elle exploite ainsi indirectement toutes les variables disponibles, et obtient un BER très prometteur de 7.5%. Il est à noter que l'on pourrait également considérer le moyennage de modèles de coclustering afin d'améliorer encore ces performances.

La participation à ce challenge a permis de confirmer le potentiel des modèles en grille sur des bases complexes. Les pistes d'amélioration les plus évidentes pour guider nos travaux futurs sont :

- adapter les algorithmes d'optimisation des modèles en grille aux très grands nombres de variables, pour d'une part obtenir de meilleures grilles, d'autre part produire plus rapidement une plus grande diversité de grilles,
- approfondir les pistes du moyennage de modèles en grille et du coclustering pour atteindre un niveau de maturité similaire à celui des modèles basés sur le classifieur Bayesien naïf,
- combiner les classifieurs basés sur les grilles et les classifieurs basés sur l'hypothèse Bayesienne naïve.

4.5.5 Predictive Uncertainty Competition

Cette section est issue en grande partie des travaux de Carine Hue sur la validation expérimentale de la méthode de discrétisation univariée pour la régression [Boullé and Hue, 2006] et sur son application en régression de rang à l'aide d'un régresseur Bayesien naïf [Hue and Boullé, 2007b, Hue and Boullé, 2007a].

La Predictive Uncertainty Competition [Cawley *et al.*, 2006], organisée au sein de la conférence IJCNN 2006, a pour but d'évaluer la capacité des méthodes de régression à effectuer des prédictions probabilistes, c'est à dire à prédire pour chaque individu en test la distribution des valeurs de la variable à régresser.

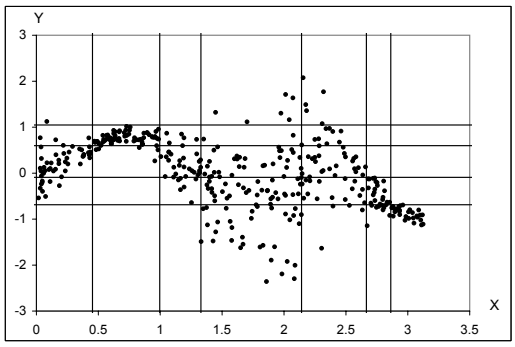
Les jeux de données sont résumés dans le tableau 4.8. Le jeu de données Synthetic est fourni à titre illustratif, avec un problème de régression univariée hétéroscédastique. Les trois autres jeux données proviennent du domaine de la météorologie.

Le critère d'évaluation est le NLPD (negative log estimated predictive density). Globalement, 9 groupes de recherche ont participé au challenge, dont les détails sont disponibles sur le site `http ://theoval.cmp.uea.ac.uk/~gcc/competition/`.

TAB. 4.8 – Jeux de données du Feature Selection Challenge.

Dataset	Variables	Individus		
		Train	Valid	Test
Synthetic	1	256	128	1024
SO2	27	15304	7652	7652
Precip	106	7031	3515	3517
Temp	106	7117	3558	3560

L'utilisation d'une grille bidimensionnelle en régression pour l'estimation de densité des rangs d'une variable numérique à expliquer, conditionnellement aux rangs d'une variable numérique explicative est illustrée sur la figure 4.18 pour le jeu de données Synthetic. Les 384 individus des bases d'apprentissage et validation réunies donnent lieu à 7 intervalles de rangs pour la variable explicative X et 5 intervalles de rangs pour la variable à expliquer Y. Pour chaque intervalle de X, les effectifs de la grille permettent de calculer directement la probabilité que le rang de Y soit dans un intervalle donné. Par exemple, pour le 1er



	P1	P2	P3	P4	P5	P6	P7	Tot.
C5	1	0	0	6	5	0	0	12
C4	4	68	5	2	9	0	0	88
C3	38	5	26	32	24	0	0	125
C2	9	0	6	33	15	21	0	84
C1	0	0	0	34	4	9	28	75
Total	52	73	37	107	57	30	28	384

FIG. 4.18 – Diagramme de dispersion, partitionnement bivarié et effectifs de la grille MODL pour le jeu de données Synthetic

intervalle de X, les rangs de Y sont distribués autour du 3^{e} intervalle de Y, alors que pour le 2^{e} intervalle de X, ils sont concentrés sur le 4^{e} intervalle de Y.

Plusieurs régresseurs univariés issus de modèles en grille bidimensionnelle peuvent être combinés en faisant l'hypothèse naïve d'indépendance, étendant ainsi le classifieur Bayesien naïf au problème de la régression de rangs.

Le critère d'évaluation du challenge est basé sur la prédiction des valeurs et non des rangs, ce qui a demandé une adaptation spécifique. Les probabilités associées aux prédictions, constantes par intervalle de rangs, ont été transformées en probabilités constantes par intervalle de valeur, en utilisant le mapping empirique entre rangs et valeurs.

Nous avons tout d'abord constaté les mauvaises performances de l'estimateur Bayesien naïf utilisant l'ensemble des prédicteurs. Lorsque l'hypothèse d'indépendance n'est pas respectée, il est connu qu'elle dégrade fortement l'estimation des probabilités a posteriori [Frank *et al.*, 1998]. Ces mauvaises performances sont donc certainement dues à des corrélations importantes entre les variables explicatives.

Nous avons effectué deux soumissions, la première univariée basée sur la meilleure variable sélectionnée selon le taux de compression du modèle en grille et la seconde bivariée basée sur les deux meilleurs variables conjointement. Le challenge étant terminé au moment où notre méthode était prête, les soumissions ont été transmises directement à Gavin Cawley, l'organisateur du challenge, qui nous a communiqué les résultats en test.

La soumission univariée termine 1re sur SO2, 3^{e} sur Precip et 7^{e} sur Temp. La soumission bivariée, toujours meilleure que l'univariée, termine 1re sur Precip.

La participation à ce challenge a permis de valider l'approche de la régression par grille. Les bonnes performances du prédicteur univarié montrent la qualité du partitionnement bidimensionnel obtenu malgré la manipulation exclusive des rangs et non des valeurs durant cette étape. D'autre part, le régresseur bivarié est toujours meilleur que l'estimateur univarié. Cela indique la présence d'informations supplémentaires et nous encourage à améliorer le régresseur Bayesien naïf par le biais d'une sélection de variables ou d'un moyennage de modèles.

4.6 Évaluation sur des données France Telecom

Cette section présente l'outil de préparation des données Khiops, issu des travaux sur les modèles en grille, et décrit l'architecture de la Plate-forme d'Analyse Client, visant à rechercher automatiquement une représentation des données performante dans les très grandes bases de données.

4.6.1 L'outil de préparation des données Khiops

L'outil de préparation des données Khiops intègre les travaux sur les modèles en grille et les diffuse dès qu'ils ont atteint une maturité suffisante. La version actuellement diffusée comprend les fonctionnalités principales suivantes :

- préparation des données :
  . discrétisation supervisée des variables numériques,
  . groupement de valeurs supervisé des variables catégorielles,
  . rapport de statistique descriptive avec classement des variables par taux de compression décroissant,
  . recodage des données,
- modélisation par classifieur Bayesien naïf :
  . prétraitement univarié par discrétisation et groupement de valeurs,
  . sélection de variables selon un critère Bayesien,
  . moyennage de modèles par taux de compression,
  . rapport de modélisation avec évaluation détaillée en test,
  . déploiement des modèles.

L'outil est écrit en langage C++ pour la partie algorithmique, et en Java pour l'interface graphique. Il est utilisable à la fois en mode interface graphique et en mode batch, ce qui permet de l'intégrer aisément dans une chaîne de traitements.

La version actuellement diffusée est utilisée en interne à France Telecom dans de nombreux domaines applicatifs :

- marketing : production de scores d'appétence à de nouveaux services télécom,
- text mining : classification de textes,
- web mining : placement de bandeaux publicitaires sur des pages web,
- étude technico-économique : estimation du prix des combinés téléphoniques en fonction de leurs caractéristiques techniques,
- trafic internet : caractérisation du type d'application et identification du trafic peer-to-peer à partir des caractéristiques des trames IP,
- ergonomie : détection d'émotion dans un dialogue,
- sociologie des usages : étude de corrélation entre données d'enquête et consommation sur la marché entreprise,
- ....

Cette version est également diffusée en externe sous forme de shareware, sur le site `http ://www.francetelecom.com/en/group/rd/offer/software/applications/providers/-khiops.html`.

Les prochaines versions de l'outil intégreront les travaux sur les grilles multivariées pour la classification supervisée et non supervisée, en cours de finalisation, et les travaux sur la régression, en cours de développement.

4.6.2 La Plate-forme d'Analyse Client

Le projet PAC (Plate-forme d'Analyse Client) est un projet mené à France Télécom R&D pour étudier l'industrialisation du processus Data Mining pour les très grandes bases de données. Notre contribution à ce projet correspond à la partie préparation des données, dont nous présentons ici l'architecture et l'évaluation.

4.6.2.1 Les systèmes d'information décisionnels

Les systèmes d'information décisionnels des grandes entreprises sont généralement structurés sur trois niveaux :

- l'entrepôt de données est une base de données relationnelle complexe et volumineuse, modélisant toutes les interactions de l'entreprise avec son environnement : clients, fournisseurs, organisation interne...,
- les datamarts utilisent des bases de données relationnelles simplifiées, en vue d'une utilisation particulière : l'analyse des produits, la gestion, l'analyse des clients, l'optimisation du réseau de télécommunications...,
- les bases d'études sont des tableaux croisés dédiés à la réalisation d'une étude, par exemple : la fidélisation, l'appétence à un produit, la détection de fraude, la détection de panne dans un réseau...

L'analyste de données intervient généralement au niveau des bases d'études, ce qui constitue un facteur limitant en raison du caractère figé des variables explicatives disponibles. En effet, si l'analyste veut construire de nouvelles variables pour améliorer les modèles, pour s'adapter à un changement d'environnement comme l'apparition d'un nouveau service, ou pour construire un nouveau score, il faut faire évoluer, voire créer une nouvelle base d'études. Le coût est important puisqu'il peut s'agir d'un projet informatique à part entière.

4.6.2.2 Architecture de préparation des données

Notre approche pour permettre la construction de scores utilisant au maximum l'information stockée dans l'entrepôt de données consiste à automatiser le processus de traitement de données et à se focaliser sur les seules données pertinentes.

L'architecture proposée se base sur les éléments suivants :

- un modèle de structuration des données,
- un langage de construction de variables,
- un outil d'extraction de représentation.

Ces trois éléments permettent une automatisation complète de la recherche de représentation dans un espace de très grande dimension.

Structuration des données. Le *Data Folder* permet de décrire formellement la structure d'un datamart sous la forme d'un schéma relationnel en étoile. Chaque individu du domaine applicatif est stocké au moyen d'une table principale et de plusieurs tables secondaires en relation 1-n. Par exemple, un client est caractérisé par les services auxquels il a souscrit, par l'historique de ses détails de communication et par l'ensemble de ses interactions avec les services après-vente de France Télécom.

Construction de variables. Le *Data Folder Query Language* (DFQL) permet de décrire de façon synthétique des requêtes de construction de variables. Par exemple, une requête "moyenne des durées d'appel sur téléphone fixe par mois (12 mois), jour de la semaine (7 jours) et type d'appel (local, national, internationnal)" permet de construire $252 = 12 * 7 * 3$ variables.

Extraction de représentation. L'outil Khiops est utilisé pour extraire les variables informatives parmi l'ensemble des variables construites à partir des requêtes DFQL.

4.6.2.3 Evaluation

Pour mesurer la fiabilité des scores produits par la Plateforme d'Analyse Client, nous avons comparé des scores construits avec ou sans notre technologie, sur un problème de prévision de la résiliation de l'abonnement au service ADSL (identification des clients "fragiles").

Un Data Folder comportant une demi-douzaine de tables secondaires autour d'une table client principale est constitué, puis alimenté à partir du système d'information décisionnelle de France Télécom. L'alimentation porte sur un ensemble de deux millions de clients sur un historique de six mois, représentant un volume de l'ordre de 100 Go. Une centaine de requêtes DFQL sont écrites et appliquées à un échantillon de 100000 clients en apprentissage, ce qui produit après construction de variables une table de données en apprentissage comportant 100000 individus et 50000 variables. L'outil Khiops, appliqué à cette table de données, sélectionne une centaine de variables, utilisées pour la modélisation. Le déploiement du modèle est automatisé en extrayant les sous-requêtes DFQL correspondant aux variables sélectionnées et en les appliquant pour extraire la représentation nécessaire à la modélisation.

La performance prédictive est évaluée en test au moyen du critère du gain à 20%, c'est à dire de la proportion de clients "fragiles" correctement identifiés quand on sélectionnne les 20% de clients ayant la plus forte probabilité prédite de fragilité. Une prédiction aléatoire obtient un gain de 20%, ce qui constitue le score plancher. La technique actuellement utilisée dans les services opérationnels se base sur 200 variables et obtient un gain de 45%, soit environ deux fois le gain aléatoire. Avec notre architecture, le gain s'élève à 65%, soit trois fois le gain aléatoire et une fois et demi celui de la technique actuellement utilisée. De plus, l'inspection des variables sélectionnées a permis de mieux caractériser les clients fragiles. Le langage DFQL s'exprimant dans les termes du domaine applicatif, les variables construites sont en effet aisément interprétables.

4.6.2.4 Bilan

La Plate-forme d'Analyse Client permet de construire des modèles de prévision basés sur un nombre de variables explicatives de deux ordres de grandeurs au-dessus de ce qui se fait actuellement. La conséquence est une nette augmentation de la qualité des modèles. Cette plate-forme repose sur une architecture novatrice permettant d'automatiser les traitements couplée avec des méthodes performantes de construction et sélection de variables. De façon indirecte, ce projet a permis de valider la tenue de charge et la robustesse des méthodes de l'outil Khiops sur de très grandes volumétries.

4.7 Conclusion

Les modèles en grille recouvrent de nombreuses problématiques de l'analyse de données, ce qui rend difficile leur évaluation exhaustive. Le choix a été fait d'inventorier les caractéristiques générales de ces modèles, au moyen d'expérimentations informatives.

L'évaluation analytique des modèles en grille permet de démontrer des propriétés intéressantes, notamment concernant les seuils d'apprentissage. Bien que non paramétrique, l'approche proposée ne nécessite que très peu d'individus pour identifier des motifs informatifs, ce qui constitue un comportement remarquable dans un cadre non asymptotique. L'évaluation analytique permet également de qualifier une limite intrinsèque de la méthode sur la complexité maximale des motifs apprennables : environ 2^K individus sont nécessaires pour apprendre un motif comportant K variables s'exprimant conjointement.

L'évaluation des prétraitements par grille en univarié et bivarié confirme les attentes théoriques et montre les nombreux atouts de ces méthodes, principalement leur simplicité de mise en oeuvre, leur excellente robustesse et performance prédictive et leur facilité d'interprétation.

L'évaluation de l'utilisation de ces prétraitements dans le cadre du classifieur Bayesien naïf montre qu'il est nécessaire d'adapter les méthodes existantes pour tirer parti des qualités des prétraitements par grille. Des améliorations du classifieur Bayesien naïf sont présentées, principalement la sélection de variables selon un critère Bayesien et le moyennage de modèles par taux de compression. Ces améliorations sont évaluées intensivement sur 30 bases de l'UCI. Les résultats démontrent une amélioration significative des performances, notamment pour la qualité de l'estimation de densité conditionnelle, habituellement médiocre dans le cadre du classifieur Bayesien naïf.

Les modèles en grille sont ensuite évalués en multivarié dans trois contextes : supervisé, non supervisé et coclustering des individus et des variables. Les modèles en grille apparaissent pertinents pour la construction automatique de variables informatives et pour leur interprétabilité. Bien que compétitifs en termes de performances prédictives, leur limite sur la complexité maximale des motifs apprenables devient une contrainte pour des bases ayant trop de variables. A nouveau, il paraît nécessaire d'adapter des méthodes existantes ou de créer de nouvelles méthodes pour tirer parti des grilles multivariées. Cette hypothèse est testée en adaptant le moyennage de modèles par taux de compression aux modèles en grille. Les évaluations démontrent un fort potentiel en performances prédictives, au prix toutefois d'un surcoût lié au moyennage pour le déploiement des modèles.

Les modèles en grille sont ensuite évalués lors de challenges internationaux, ce qui permet de les confronter aux meilleures méthodes de l'état de l'art sur des jeux de données complexes. Ces évaluations ont confirmé la robustesse des modèles en grille et ont facilité l'identification d'axes d'amélioration permettant de les rendre compétitifs en termes de performances prédictives.

Enfin, l'application des modèles aux données France Télécom est présentée. L'outil Khiops, qui industrialise les méthodes issues des travaux sur les modèles en grille, est utilisé dans de nombreux domaines applicatifs. Il est notamment utilisé dans le domaine marketing, sur des échantillons de données dont la volumétrie atteint environ 100000 individus pour 50000 variables.

De façon synthétique, les modèles en grille sont particulièrement adaptés à la préparation des données pour le data mining, en permettant une identification automatique, rapide, fiable, précise et interprétable des motifs informatifs sans faire aucune hypothèse préalable sur les données. En phase de modélisation, l'utilisation efficace de ces prétraitements requiert la mise au point de nouvelles méthodes d'apprentissage. Parmi celles-ci, le moyennage de modèles et le contournement de l'hypothèse d'indépendance du classifieur Bayesien naïf semblent deux voies prometteuses.

5

Positionnement de l'approche

Sommaire

Après avoir introduit les modèles en grille et leur critère d'évaluation dans le chapitre 2, résumé leur algorithme d'optimisation dans le chapitre 3, et présenté leur évaluation dans le chapitre 4, nous proposons ici un positionnement de l'approche MODL.

Les modèles en grille sont applicables à de nombreux problèmes de l'analyse de données, pour le prétraitement des données, la classification supervisée, la régression, la classification non supervisée, l'estimation de densité, et il n'est pas possible de viser l'exhaustivité pour un positionnement. Notre choix est de réutiliser la démarche qui a conduit à l'élaboration de cette famille de modèles, en partant des cas les plus simples, discrétisation et groupement de valeurs supervisé, puis en étendant l'approche au cas d'une variable à expliquer numérique, enfin en généralisant au cas multivarié.

La section 5.1 dresse une typologie des méthodes de discrétisation supervisée de l'état de l'art, avant d'exposer les caractéristiques de la méthode de discrétisation MODL.

La section 5.2 dresse un panorama des approches de groupement de valeurs supervisé, puis positionne notre approche.

La section 5.3 propose une catégorisation des méthodes de régression en régression de valeurs, de quantiles ou de rangs, puis classe notre approche en tant que régression probabiliste de rangs.

La section 5.4 présente dans le cas multivarié un éventail de méthodes comparables aux modèles en grille, introduites dans de nombreux domaines de l'analyse de données et exploitant une large variété d'approches, puis synthétise les apports de notre méthode.

La section 5.5 analyse la démarche de modélisation utilisée pour l'élaboration des modèles en grille. Après avoir rappelé les objectifs de modélisation, puis dégagé les contraintes liées à ces objectifs, les choix de modélisation sont présentés et discutés de façon comparative, notamment par rapport aux approches Bayesienne et de principle de description de longueur minimum pour la sélection de modèles.

Enfin, la section 5.6 conclut ce chapitre.

5.1 Discrétisation supervisée univariée

Après avoir introduit le problème de la discrétisation supervisée, on présente une typologie des méthodes existantes, illustrée par quelques méthodes représentatives, avant de positionner notre approche.

5.1.1 Introduction

La discrétisation des variables numériques est un sujet largement traité dans la bibliographie [Catlett, 1991, Holte, 1993, Dougherty *et al.*, 1995, Zighed and Rakotomalala, 2000, Liu *et al.*, 2002]. De nombreuses méthodes d'apprentissage sont basées sur le traitement des variables catégorielles uniquement. Il est donc nécessaire de discrétiser les variables numériques, c'est à dire de découper leur domaine en un nombre fini d'intervalles. Par exemple, les arbres de décision discrétisent les variables numériques avant de sélectionner la variable de décision à chaque noeud de l'arbre. Les algorithmes à base d'ensembles de règles exploitent une méthode de discrétisation afin de produire des règles concises et interprétables. Les réseaux Bayesiens ont besoin de variables discrétisées pour calculer les tables de probabilités conditionnelles.

C'est certainement dans le contexte des arbres de décision que les méthodes de discrétisation ont été le plus étudiées. C4.5 [Quinlan, 1986, Quinlan, 1993] utilise le gain

informationnel basé sur l'entropie de Shannon, CART [Breiman *et al.*, 1984] utilise l'indice de Gini (une mesure d'impureté des intervalles), CHAID [Kass, 1980] s'appuie sur une méthode de type ChiMerge [Kerber, 1991] à base de test du χ^2, SIPINA [Zighed and Rakotomalala, 1996] utilise le critère Fusinter [Zighed *et al.*, 1998] basé sur des mesures d'incertitude sensibles aux effectifs.

Le problème de la discrétisation est un problème de compromis entre qualité informationnelle (intervalles homogènes vis à vis de la variable à prédire) et qualité statistique (effectif suffisant dans chaque intervalle pour assurer une généralisation efficace). De façon générale, une méthode de discrétisation est définie par un critère d'évaluation, pour mesurer la qualité d'une discrétisation, un algorithme d'optimisation, pour rechercher une discrétisation performante, et un critère d'arrêt, pour stopper l'algorithme d'optimisation.

5.1.2 Les critères d'évaluation

On présente ci-dessous une typologie des critères de discrétisation supervisée.

5.1.2.1 Supervisé versus non supervisé.

Les critères cités précédemment dans le cadre des arbres de décision sont tous supervisés. Ils tiennent compte de la variable à expliquer pour évaluer une discrétisation, alors que les critères non supervisés n'utilisent que la variable explicative. Les méthodes non supervisées les plus classiques sont EqualWidth et EqualFrequency. La méthode EqualWidth divise le domaine numérique en intervalles de largeur égale, alors que la méthode EqualFrequency divise l'ensemble des individus en intervalles d'effectif égal.

Dans la plupart des articles se comparant à ces méthodes, le nombre d'intervalles est fixé à 10. Certains auteurs proposent des choix heuristiques, comme le log du nombre de valeurs numériques distinctes [Dougherty *et al.*, 1995]. Il est à noter que le problème du choix du nombre d'intervalles pour la méthode EqualWidth est un sujet d'étude à part entière dans la communauté scientifique de l'estimation de densité par histogrammes [Sturges, 1926, Scott, 1979, Castellan, 1999, Birgé and Rozenholc, 2002].

Remarque 5.1. *Les critères supervisés sont préférables dès lors qu'il s'agit de modéliser la distribution de probabilité conditionnelle de la variable à expliquer. Ils ne sont pas limités dans leur expressivité comme les critères non supervisés, pour lesquels le choix de l'emplacement des bornes de discrétisation est fortement contraint et ne peut pas tenir compte des variations de densité conditionnelle.*

5.1.2.2 Global versus local.

Les critères globaux recherchent une partition de tous les individus alors que les critères locaux ne s'intéressent qu'à un sous-ensemble des individus. La distinction "global versus local", aussi appelée "statique versus dynamique" dans la littérature, est essentiellement liée à l'utilisation des méthodes dans le cadre des arbres de décision, qui peuvent soit discrétiser les variables numériques une fois pour toutes avant la construction de l'arbre, soit discrétiser les variables localement à chaque noeud de l'arbre, sur une sous-population d'individus.

5.1.2.3 Binaire versus n-aire.

Les critères binaires recherchent la meilleure discrétisation en deux intervalles, alors que les critères n-aires recherchent une discrétisation en un nombre quelconque d'intervalles. Il est à noter que les critères binaires sont souvent utilisés en conjonction avec un algorithme récursif pour produire des discrétisations n-aires.

Remarque 5.2. *En raison de leur forte expressivité, les critères n-aire sont préférables pour modéliser la distribution de probabilité conditionnelle de la variable à expliquer.*

5.1.2.4 Familles de critères

Plusieurs familles de critères ont été étudiées dans la littérature. Les principales sont basées sur :

- le taux d'erreur : il s'agit de minimiser le taux d'erreur en apprentissage,
- l'indépendance entre intervalles : il s'agit de fusionner les intervalles adjacents ayant une distribution similaire des valeurs à expliquer,
- évaluation de l'entropie : il s'agit d'évaluer l'entropie conditionnelle de la variable à expliquer.

Les critères basés sur le taux d'erreur sont adaptés à la modélisation déterministe, alors que ceux basés sur l'indépendance des distributions entre intervalles ou sur l'entropie conditionnelle concernent la modélisation probabiliste.

5.1.2.5 Pénalisation des critères

La discrétisation supervisée est un problème de compromis entre finesse et fiabilité. Les critères de discrétisation sensibles à la finesse des informations produisent généralement trop d'intervalles, ce qui nuit à leur fiabilité.

Pour améliorer la fiabilité des critères, certaines méthodes pénalisent explicitement les discrétisations comportant de nombreux intervalles ou ayant des intervalles de faible effectif. Cette pénalisation est incorporée soit dans le critère directement, par exemple par un ratio dont le dénominateur est le nombre d'intervalles, soit sous forme de paramètres utilisateurs. Des paramètres sont également utilisés dans les critères basés sur une approximation, en imposant par exemple un effectif minimum pour fiabiliser l'estimation empirique des probabilités, ou dans les critères basés sur un test statistique, pour fixer un seuil de décision. Une autre approche pour obtenir un critère d'évaluation fiable est de poser le problème de la discrétisation comme un problème de sélection de modèles, en adoptant par exemple une approche Bayesienne MAP, une approche MDL [Rissanen, 1978], ou un test statistique dans le cas d'un choix entre deux hypothèses.

De façon synthétique, on peut retenir les caractéristiques suivantes (non exclusives) employées pour la fiabilisation des critères :

- pénalisation en fonction du nombre d'intervalles,
- pénalisation en fonction des effectifs par intervalle,
- utilisation d'une approche de sélection de modèles,
- utilisation de paramètres.

Remarque 5.3. *La pénalisation explicite des discrétisations comportant trop d'intervalles ou contenant des intervalles de faible effectif ne garantit ni la finesse, ni la fiabilité des discrétisations.*

Remarque 5.4. *Les critères binaires résolvent le problème de fiabilité en se limitant à deux intervalles. S'ils sont utilisés récursivement pour produire des discrétisations n-aires, la fiabilité redevient un problème, notamment en raison de l'application du même critère de nombreuses fois, sur des sous-échantillons de tailles très variées.*

Remarque 5.5. *Les critères incorporant un paramétrage utilisateur ne sont pas adaptés à l'automatisation de la phase de préparation des données du Data Mining, puisqu'ils réclament une intervention humaine. Même quand les paramètres sont fixés de façon interne à la méthode, ils résultent souvent de compromis heuristiques adaptés à certaines situations, ce qui n'est pas satisfaisant pour traiter de façon générique, fiable et fine tout problème d'analyse de données.*

5.1.3 Les algorithmes d'optimisation

On présente ci-dessous les principaux algorithmes de discrétisation. Pour les complexités algorithmiques annoncées, on distingue le cas des critères additifs, permettant des implémentations optimisées, du cas général, pour des critères nécessitant une évaluation sur tous les intervalles.

5.1.3.1 Heuristique gloutonne ascendante

L'heuristique gloutonne ascendante part d'une discrétisation maximale contenant autant d'intervalles élémentaires que d'individus et évalue toutes les fusions d'intervalles adjacents. La fusion qui produit le meilleur critère est effectuée s'il y a amélioration (ou en fonction des conditions d'arrêt), et le processus est répété tant qu'il y a amélioration.

Cette heuristique a une complexité algorithmique de $\mathrm{O}(N^3)$ dans le cas général, puisqu'à chacune des $\mathrm{O}(N)$ étapes de fusion, $\mathrm{O}(N)$ fusions d'intervalles doivent être évaluées, chacune reposant sur un critère portant sur $\mathrm{O}(N)$ intervalles. Néanmoins, pour les critère additifs, les algorithmes sont implémentables en $\mathrm{O}(N \log N)$.

Remarque 5.6. *L'heuristique gloutonne ascendante a tendance à construire des discrétisations comportant trop d'intervalles, puisqu'elle part d'une discrétisation contenant un nombre maximal d'intervalles et s'arrête au premier optimum local rencontré. Cela aboutit à des discrétisations informatives mais peu robustes.*

5.1.3.2 Heuristique gloutonne descendante

L'heuristique gloutonne descendante part d'une discrétisation minimale en un seul intervalle et évalue toutes les coupures en deux sous-intervalles. La coupure qui produit le meilleur critère est effectuée s'il y a amélioration (ou en fonction des conditions d'arrêt), et le processus est appliqué récursivement aux sous-intervalles tant qu'il y a amélioration.

Comme dans le cas ascendant, la complexité est de $\mathrm{O}(N^3)$ dans le cas général. A notre connaissance, même pour des critères additifs, cette complexité ne peut être meilleure que $\mathrm{O}(N^2)$ dans le pire des cas (quand les décisions de coupures se produisent $\mathrm{O}(N)$ fois dans le sous-intervalle le plus peuplé, d'effectif $\mathrm{O}(N)$). En pratique, la complexité moyenne est toutefois de $\mathrm{O}(N \log N)$.

Remarque 5.7. *L'heuristique gloutonne descendante a tendance à construire des discrétisations comportant trop peu d'intervalles, puisqu'elle part d'une discrétisation contenant un unique intervalle et s'arrête au premier optimum local rencontré. Cela aboutit à des discrétisations robustes mais peu informatives.*

5.1.3.3 Algorithme optimal

Pour un critère binaire, il est aisé de construire la discrétisation optimale en deux sous-intervalles, par examen exhaustif de tous les points de coupure potentiels, en $O(N)$ pour les critères additifs, $O(N^2)$ sinon.

Pour un critère n-aire additif, il est possible de trouver la discrétisation optimale en $O(N^3)$ au moyen d'un algorithme basé sur la programmation dynamique.

Remarque 5.8. *Quand un critère binaire est utilisé récursivement dans une heuristique gloutonne ascendante ou descendante, on aboutit à une discrétisation n-aire. Néanmoins, la notion de discrétisation optimale n'a pas de sens dans ce cas, puisque le critère binaire ne permet pas de comparer les discrétisations n-aires entre elles.*

Remarque 5.9. *L'intérêt de l'algorithme optimal est essentiellement théorique, pour évaluer la performance des heuristiques d'optimisation. Sa complexité algorithmique en $O(N^3)$ ne permet pas son utilisation en pratique dans le cas de grandes bases de données.*

5.1.4 Quelques méthodes représentatives

Le problème de la discrétisation supervisée a été abondamment traité dans la littérature, avec plusieurs centaines de contributions. On se limite ici à la présentation de quelques méthodes représentatives pour les familles de critères les plus usuelles.

5.1.4.1 Méthodes basées sur le taux d'erreur

Pour les méthodes optimisant le taux d'erreur, le sur-apprentissage est généralement contrôlé en imposant un nombre maximal d'intervalles ou un effectif minimum par intervalle. Dans [Holte, 1993], l'effectif minimal par intervalle est fixé à 6 de façon heuristique. Dans [Maass, 1994], où l'algorithme optimal pour le critère de minimisation du taux d'erreur en apprentissage est présenté, le nombre maximal d'intervalles est un paramètre utilisateur. Dans [Pfahringer, 1995], la méthode MDL-DISC utilise le principe MDL [Rissanen, 1978] pour coder la valeur majoritaire et les individus mal classés dans chaque intervalle. Le critère proposé n'est pas additif, et l'auteur présente une heuristique d'optimisation se limitant à une recherche heuristique dans un ensemble de discrétisations basé sur 32 intervalles élémentaires. Ces intervalles élémentaires proviennent de l'application préalable d'une version simplifiée de la méthode descendante D2 [Catlett, 1991]. Dans [Kurgan and Cios, 2004], la méthode CAIM utilise pour chaque intervalle un ratio entre l'effectif de la valeur à expliquer majoritaire et l'effectif de l'intervalle. Pour éviter de produire trop d'intervalles, le critère est divisé par le nombre d'individus.

Remarque 5.10. *Les méthodes basées sur le taux d'erreur se focalisent uniquement sur la valeur à expliquer majoritaire, en ignorant l'ensemble de la distribution des valeurs à expliquer, ce qui*

limite leur performance prédictive [Kohavi and Sahami, 1996]. Etant plus adapté à la modélisation déterministe qu'à la modélisation probabiliste, leur intérêt est limité en préparation des données.

5.1.4.2 Méthodes basées sur le critère du χ^2

Le critère du χ^2 est souvent utilisé comme mesure de dépendance des distributions des valeurs à expliquer entre intervalles. Deux intervalles adjacents présentant des différences de distribution statistiquement significatives sont séparés, sinon ils sont fusionnés. Le χ^2 est exploité en tant que critère de discrétisation binaire dans [Bertier and Bouroche, 1981] avec une heuristique gloutonne descendante et dans [Kerber, 1991] avec une heuristique gloutonne ascendante. Pour ces deux méthodes, le seuil de rejet de l'hypothèse d'indépendance est un paramètre utilisateur.

Dans [Lechevallier, 1990], le critère du χ^2 est optimisé sur l'ensemble de tous les intervalles pour produire des discrétisations n-aire au moyen d'un algorithme optimal. Afin de trouver automatiquement le bon nombre d'intervalles, l'auteur propose d'optimiser une version normalisée du χ^2, comme par exemple l'intensité de dépendance ϕ^2 de Pearson, le coefficient V de Cramer ou le coefficient T de Tschuprow (cf. figure 5.1 pour la définition de ces différents coefficients), dans le but de permettre des comparaisons équitables d'un point de vue numérique entre discrétisations d'arités différentes.

Dans [Boullé, 2004a], le critère d'évaluation d'une discrétisation est le niveau de confiance associé au test du χ^2, ce qui permet des comparaisons équitables d'un point de vue statistique, pas seulement numérique. La méthode d'optimisation est l'heuristique gloutonne ascendante, qui a tendance à produire trop d'intervalles. Pour remédier à cet inconvénient, l'algorithme prend en compte une contrainte d'effectif minimum par intervalle égal à $\sqrt{N}$. Cette contrainte d'effectif minimum empêche la détection de motifs trop fins sans fournir de vraie garantie de robustesse. Dans [Boullé, 2003], la contrainte d'effectif minimum par intervalle est remplacée par un test statistique sur les variations maximales du critère du χ^2 lors de l'algorithme glouton ascendant. Cela permet de garantir pour une probabilité utilisateur donnée que toute variable explicative indépendante de la variable à expliquer sera discrétisée en un seul intervalle. Cette méthode s'avère compétitive, mais reste limitée par la nécessité d'un paramètre utilisateur et par les conditions d'application du test du χ^2 dans les cas non asymptotiques (comme pour toute méthode exploitant le test du χ^2).

Remarque 5.11. *Quand le critère du χ^2 est utilisé directement comme critère d'évaluation, son calcul se fait de façon exacte par comptage. Par contre, ce critère ne permet pas une comparaison équitable entre discrétisations d'arités différentes. Les coefficients normalisés basés sur une transformation du χ^2 permettent d'uniformiser l'étendue des valeurs du critère. Néanmoins, cette normalisation adhoc ne tient pas compte de la nature des variables, ce qui ne permet qu'une comparaison numériquement équitable, pas une comparaison statistiquement équitable. Ceci est vérifié expérimentalement dans le cas du critère de Tschuprow dans [Boullé, 2005a].*

Remarque 5.12. *Quand c'est le niveau de confiance associé au test du χ^2 qui est optimisé, on s'approche d'une comparaison statistiquement équitable, puisque c'est toujours une probabilité qui est l'objet de la comparaison. Par contre, le critère ne résulte plus d'un calcul exact, en étant basé sur une approximation par la loi Gamma, valide uniquement asymptotiquement. Certains auteurs comme [Cochran, 1954, Connor-Linton, 2003] recommandent un effectif théorique*

minimum de 5 par case du tableau de contingence. De façon générale, une contrainte d'effectif minimal par intervalle doit être intégrée pour trouver un compromis entre finesse et fiabilité des méthodes, ce qui empêche leur applicabilité dans le cas des échantillons de petite taille ou de distribution fortement déséquilibrée des valeurs à expliquer.

5.1.4.3 Méthodes basées sur l'entropie

L'entropie conditionnelle est à la base de nombreuses méthodes de discrétisation pour évaluer la "pureté" des intervalles vis à vis des valeurs de la variable à expliquer. L'optimisation directe de ce critère sur l'ensemble des intervalles conduit à autant d'intervalles que de valeurs, ce qui est optimal pour l'entropie, mais mauvais en généralisation.

Dans les arbres de décision ID3 [Quinlan, 1986] et C4.5 [Quinlan, 1993], l'entropie est à la base d'un critère binaire optimisé tel quel, ce qui évite le sur-apprentissage en se limitant aux discrétisations en deux intervalles. Dans la méthode de discrétisation D2 [Catlett, 1991], le critère d'entropie est appliqué récursivement avec l'heuristique gloutonne descendante, en arrêtant l'algorithme si l'un des critères suivants est activé : l'effectif d'un intervalle est inférieur à 14, le nombre total d'intervalles dépasse 8, l'amélioration du critère est la même sur tous les points de coupure, ou tous les individus d'un intervalle ont la même valeur à expliquer.

Dans la méthode BalancedGain [Kononenko *et al.*, 1984], l'entropie de la discrétisation est divisée par le logarithme du nombre d'intervalles pour pénaliser les partitions trop fines. Cette méthode est évaluée au moyen de l'algorithme optimal de discrétisation dans [Elomaa and Rousu, 1999].

Dans la méthode Fusinter [Zighed *et al.*, 1998], le critère est basé sur une mesure d'incertitude, l'entropie quadratique, et les intervalles de petits effectifs sont pénalisés par un terme additif inversement proportionnel aux effectifs. La méthode nécessite deux paramètres utilisateurs, le premier pour l'estimation empirique des probabilités conditionnelles, le second pour pondérer le terme de pénalisation.

Dans la méthode MDLPC [Fayyad and Irani, 1992], les auteurs posent le problème de la discrétisation binaire comme un problème de sélection entre deux hypothèses : couper ou ne pas couper l'intervalle en deux sous-intervalles. Ce problème est résolu au moyen d'une approche MDL [Rissanen, 1978], ce qui fournit un critère ayant un terme de pénalisation pour le codage du point de coupure et un terme de type entropie pour le codage des données connaissant l'hypothèse de coupure.

Dans [Munteanu, 1996], une approche similaire à celle de MDLPC est adoptée en utilisant une approche Bayesienne de la sélection de modèles pour le problème de la discrétisation binaire dans le cas de deux valeurs à expliquer. Cela conduit comme précédemment à un terme de pénalisation relatif à la distribution à priori des hypothèses de coupure et un terme d'entropie relatif à la vraisemblance des données.

Remarque 5.13. *Le critère d'entropie [Shannon, 1948] est un critère valide asymptotiquement, difficile à évaluer de façon fiable en pratique [Guyon and Elisseeff, 2003]. Ce critère est toujours évalué sur la base des probabilités empiriques, généralement au moyen de l'estimateur de Laplace, ce qui pose des problèmes quand les effectifs sont insuffisants.*

TAB. 5.1 – Caractéristiques des méthodes de discrétisation de l'état de l'art comparativement à la méthode MODL.

Méthode	Critère										Algorithme			
	Nature		Arité		Famille			Pénalisation						
	Non supervisé	Supervisé	Binaire	N-aire	Taux d'erreur	Indépendance	Entropie	Effectif minimum	Nombre d'intervalles	Sélection de modèles	Paramètres	Critère additif	Optimal	Descendant	Ascendant
EqualWidth	X			X					X		X	X	X		
EqualFrequency	X			X					X		X		X		
CHAID [Kass, 1980]		X	X			X			X	X	X	X			X
ChiSplit [Bertier and Bouroche, 1981]		X	X			X			X	X	X	X		X	
BalancedGain [Kononenko *et al.*, 1984]		X		X			X		X			X		X	
CART [Breiman *et al.*, 1984]		X	X				X		X			X		X	
Tschuprow [Lechevallier, 1990]		X		X			X					X	X		
D2 [Catlett, 1991]		X	X				X	X	X			X		X	
ChiMerge [Kerber, 1991]		X	X			X			X	X	X	X			X
MDLPC [Fayyad and Irani, 1992]		X	X				X		X	X		X		X	
1R [Holte, 1993]		X		X				X				X			
C4.5 [Quinlan, 1993]		X	X				X		X			X		X	
[Maass, 1994]		X		X					X		X	X	X		
MDL-DISC [Pfahringer, 1995]		X		X				X	X	X				X	
[Munteanu, 1996]		X	X				X		X	X		X		X	
Fusinter [Zighed *et al.*, 1998]		X		X			X		X		X	X			X
BalancedGain [Elomaa and Rousu, 1999]		X		X			X		X			X	X		
Khiops [Boullé, 2004a]		X		X		X		X				X			X
Khiops [Boullé, 2003]		X		X		X			X	X	X	X			X
CAIM [Kurgan and Cios, 2004]		X		X	X				X			X		X	
MODL	X			X		X	X			X		X			X

5.1.5 Positionnement de notre approche

Comparé au problème de la modélisation prédictive dans son cadre multivarié le plus général, le problème de la discrétisation est un problème simple, souvent considéré au mieux comme une étape de prétraitement nécessaire en préalable des méthodes d'apprentissage. Par exemple, la majeure partie des articles ignore l'évaluation des méthodes de discrétisation intrinsèquement, en se concentrant sur leur utilisation dans des classifieurs plus sophistiqués, comme les arbres de décision ou les classifieurs Bayesiens naïfs.

L'originalité de la méthode de discrétisation supervisée MODL résulte essentiellement du choix de traiter explicitement la discrétisation comme un problème de modélisation à part entière, même s'il est réduit à une seule variable explicative numérique et une seule variable à expliquer catégorielle. La méthode MODL se caractérise par les éléments suivants :

- formalisation explicite d'une famille de modèles de discrétisation supervisée n-aire,
  - une discrétisation est considérée comme un modèle d'estimation de densité conditionnelle, constant par morceaux,
- adoption d'une approche Bayesienne de la sélection de modèle,
  - distribution a priori des modèles guidée par la hiérarchie des paramètres,
  - estimation des probabilités a priori et a posteriori par comptage,
  - hypothèse d'indépendance entre les intervalles,
- optimisation poussée,
  - optimisation par heuristique gloutonne ascendante, forcée jusqu'à obtenir un seul intervalle,
  - post-optimisation dans le voisinage de la meilleure discrétisation.

La littérature sur la discrétisation supervisée est très vaste. Pour chacun des éléments caractéristiques de la méthode MODL, il existe des méthodes alternatives ayant fait des choix similaires, comme le montre le tableau 5.1. La différenciation provient de l'ensemble des éléments considérés conjointement.

L'objectif étant d'estimer la densité conditionnelle de la variable à expliquer, nous avons choisi un critère supervisé n-aire, plus adapté qu'un critère non supervisé ou binaire, en raison de sa grande expressivité. Le critère MODL s'apparente aux critères de type indépendance, puisqu'il intègre cette hypothèse dans l'a priori sur les paramètres de modélisation. Il est également proche des critères de type entropie, puisque le terme de vraisemblance dans le critère MODL s'approche asymptotiquement d'une entropie.

L'approche Bayesienne utilisée pour la sélection du modèle de discrétisation ne nécessite aucun paramètre utilisateur ni aucune contrainte sur le nombre d'intervalles ou l'effectif minimum par intervalle. De plus, aucune hypothèse de validité asymptotique n'est ici nécessaire, contrairement par exemple aux méthodes utilisant le critère du χ^2, dont la loi est approximée asymptotiquement par la loi Gamma, ou le critère d'entropie, valide asymptotiquement.

Pour l'optimisation, nous avons retenu l'heuristique gloutonne ascendante de discrétisation, puisqu'elle seule garantit une complexité algorithmique en $O(N \log N)$ dans le pire des cas. De plus la méthode MODL incorpore des heuristiques de post-optimisation performantes, dans la limite de la complexité algorithmique $O(N \log N)$. Ces post-optimisations permettent une amélioration notable de la fiabilité et de la finesse des discrétisations.

La combinaison des choix de modélisation conduit à une méthode particulièrement adaptée à la préparation des données, comme l'indique le bilan (en section 4.2.5) de l'évaluation comparative des méthodes de prétraitement MODL.

5.2 Groupement de valeur supervisé univarié

Le groupement de valeurs supervisé est nettement moins étudié dans la bibliographie que la discrétisation supervisée. Cette section dresse un panorama des approches principales, puis présente le positionnement de notre approche.

5.2.1 Panorama des méthodes de groupement de valeurs

Le problème du groupement des valeurs d'une variable catégorielle consiste à partitionner l'ensemble des valeurs de la variable en un nombre fini de groupes. La plupart des modèles prédictifs à base d'arbres de décision utilisent une méthode de groupement de valeurs pour traiter les variables catégorielles, de façon à lutter contre la fragmentation des données. Les méthodes à base de réseaux de neurones n'utilisant que des données numériques ont souvent recours à un codage disjonctif complet des variables catégorielles. Dans le cas où les valeurs sont trop nombreuses, il est nécessaire de procéder au préalable à des groupements de valeurs. Ce problème se rencontre également dans le cas des réseaux Bayesiens ou de la régression logistique. De façon générale, le groupement de valeurs est une technique intéressante de préparation des données pour le Data Mining, qui permet d'identifier les groupes de valeurs homogènes vis à vis de la variable à expliquer.

Les méthodes de groupement peuvent se catégoriser en fonction de la stratégie de recherche du meilleur groupement et du type de critère d'évaluation à optimiser. Plusieurs stratégies de groupement de valeurs ont été explorées dans la bibliographie.

5.2.1.1 Les principales approches

L'approche la plus simple est la binarisation où une valeur est isolée contre toutes les autres. Une stratégie plus élaborée consiste à chercher un groupement binaire, en deux groupes de valeurs. L'algorithme Sequential Forward Selection inspiré de [Cestnik *et al.*, 1987] et évalué dans [Berckman, 1995] est un algorithme glouton qui recherche la meilleure bipartition des valeurs en déplaçant les valeurs une à une d'un premier groupe initialement complet vers un second groupe initialement vide. Dans le cas d'une variable booléenne à expliquer, [Breiman *et al.*, 1984] présentent un algorithme optimal de groupement binaire des valeurs en deux parties pour certaines familles de critère. Cet algorithme est basé sur un tri préalable des valeurs par proportion croissante de la première valeur à expliquer, puis sur le choix d'une coupure entre deux valeurs adjacentes dans cette liste triée de valeurs. La complexité de cet algorithme est en $O(V \log V)$ où V est le nombre de valeurs explicatives initiales.

L'approche la plus générale est la recherche d'un groupement n-aire, en un nombre quelconque de groupes. En se basant sur les idées de [Lechevallier, 1990, Fulton *et al.*, 1995], il est possible d'étendre le résultat d'optimalité de [Breiman *et al.*, 1984] du cas

binaire au cas n-aire à I groupes dans le cas d'une variable booléenne à expliquer, en utilisant un algorithme de programmation dynamique de complexité cubique par rapport au nombre V de valeurs explicatives initiales. Dans le cas général, il n'existe pas d'algorithme de recherche de groupement optimal autre que la recherche exhaustive, qui n'est pas envisageable. [Chou, 1991] a néanmoins mis en évidence des conditions d'optimalité permettant de réduire l'espace de recherche, et proposé un algorithme de type K-moyennes permettant de trouver une partition des V valeurs en I groupes localement optimale. La complexité algorithmique est en $O(I * V)$ multiplié par le nombre d'itérations (en général faible), mais l'optimalité globale n'est pas assurée et le nombre de groupes est un paramètre utilisateur.

En pratique, la stratégie de groupement des valeurs explicatives repose souvent sur l'utilisation d'un algorithme glouton ascendant [Kass, 1980, Quinlan, 1993]. Cet algorithme est similaire à une classification hiérarchique ascendante des valeurs et regroupe itérativement les valeurs pour optimiser un critère de qualité du groupement, en s'arrêtant quand le maximum est atteint (ce qui détermine automatiquement le nombre de groupes). Dans [Ritschard *et al.*, 2001], cet algorithme glouton est comparé avec une recherche exhaustive optimale dans le cas du critère de Tschuprow (cf. figure 5.1), et appliqué à des jeux d'essai artificiels de petite taille, dans le cas des groupements de lignes et de colonnes d'un tableau de contingence. Les résultats montrent que l'algorithme glouton trouve des solutions proches de la solution optimale.

5.2.1.2 Les critères d'évaluation

Les critères utilisés pour évaluer la qualité d'un groupement de valeurs sont très nombreux : il s'agit en fait des critères utilisés pour évaluer un tableau de contingence, analogues à ceux utilisés dans le cas de la discrétisation supervisée.

La méthode ID3 [Quinlan, 1986] utilise le gain informationnel basé sur l'entropie de Shannon pour comparer l'importance prédictive des variables, sans procéder à des groupements de valeurs. Ce critère favorisant les variables ayant de nombreuses valeurs, [Quinlan, 1993] a apporté un correctif heuristique au gain informationnel, le "gain ratio", en divisant le gain informationnel par la quantité d'information contenue dans la variable catégorielle explicative, ce qui s'apparente à une pénalisation par le nombre de valeurs explicatives.

La méthode CART [Breiman *et al.*, 1984] recherche pour chaque variable une bipartition des valeurs en utilisant l'indice de Gini. Dans le cas d'une variable à expliquer booléenne, l'algorithme permet de trouver la bipartition optimale. Dans le cas général, l'algorithme utilise une méthode de recherche exhaustive en évaluant toutes les bipartitions des valeurs explicatives pour l'indice de Gini portant sur les J valeurs à expliquer. Un critère alternatif appelé critère Twoing est également proposé, pour lequel toutes les bipartitions des valeurs à expliquer sont envisagées, et pour chacune d'entre elles, la meilleure bipartition des valeurs explicatives est recherchée en se ramenant à l'indice de Gini portant sur deux valeurs à expliquer. La complexité de cette recherche (optimale) étant exponentielle en fonction du nombre de valeurs, cette méthode n'est envisageable que dans le cas où il y a peu de valeurs explicatives ou à expliquer.

La méthode CHAID [Kass, 1980] utilise une critère de groupement des valeurs apparentée à ChiMerge [Kerber, 1991]. Il s'agit de rechercher la meilleure fusion de valeurs

en minimisant le critère du χ^2 local aux deux valeurs explicatives à fusionner, de façon à favoriser le groupement de valeurs ayant un comportement statistique similaire.

Le problème du groupement de valeurs est souvent modélisé comme un problème de réduction du tableau de contingence entre les valeurs explicatives et les valeurs à expliquer. Dans ce contexte, l'utilisation du critère du χ^2 est envisagée pour l'évaluation globale du tableau de contingence et non de façon locale à deux lignes de ce tableau de contingence comme dans CHAID. La figure 5.1, extraite de [Ritschard *et al.*, 2001], présente une série d'indicateurs normalisés usuels d'évaluation d'un tableau de contingence. Par exemple, les coefficients de Cramer ou de Tschuprow permettant une normalisation numérique de la valeur du χ^2 ont été utilisés comme critère de groupement à optimiser.

Dans [Ritschard, 2003], le tableau de contingence est évalué au moyen du rapport de vraisemblance G^2, aussi appelé déviance (cf. figure 5.1). Afin de rechercher un compromis entre information et robustesse, la méthode proposée utilise le critère BIC [Schwartz, 1978] pour pénaliser les nombres de groupes élevés. Le terme de pénalisation, en relation directe avec la réduction du nombre de cases du tableau de contingence, ne tient pas compte de la nature numérique ou catégorielle de la variable explicative.

La méthode Khiops [Boullé, 2004c] utilise le niveau de confiance associé au test du χ^2 pour évaluer les tableaux de contingence et propose un contrôle statistique de l'algorithme de groupement permettant de fiabiliser la qualité prédictive des groupements.

Le tableau 5.2 synthétise les principales approches présentées. Pour une description approfondie des méthodes à base d'arbre ou de graphe d'induction et de la façon dont elles traitent les variables catégorielles, on peut se référer à [Zighed and Rakotomalala, 2000].

5.2.2 Positionnement de notre approche

La méthode de groupement de valeurs MODL utilise exactement la même démarche que dans le cas de la discrétisation, en considérant le problème comme celui de la sélection de modèle dans un espace d'estimateurs non paramétriques de probabilité conditionnelle. La seule différence provient des caractéristiques de la variable explicative, numérique pour la discrétisation et catégorielle pour le groupement de valeurs. Dans les deux cas, on recherche une partition des valeurs explicatives, et on associe à chaque partie une distribution des valeurs à expliquer. Dans le cas d'une variable numérique, on contraint la partition recherchée à respecter la relation d'ordre entre les valeurs à expliquer, ce qui conduit à une partition en intervalles. Dans le cas d'une variable catégorielle, on envisage toutes les partitions de valeurs possibles. Ceci a des conséquences sur l'espace des modèles, le choix de la distribution a priori des modèles et les algorithmes d'optimisation.

La famille de modèles considère toutes les partitions des valeurs explicatives en groupes. Comme dans le cas de la discrétisation, l'a priori sur le choix de la partition est uniforme pour une taille de partition donnée. Les partitions étant plus nombreuses, leur dénombrement passe d'un terme de type nombre de combinaisons dans le cas des partitions en intervalles de valeurs à un terme de type nombre de Bell dans le cas des partitions en groupes de valeurs. Cela a pour conséquence de davantage pénaliser les partitions de taille importante. L'heuristique gloutonne ascendante d'optimisation nécessite également

TAB. 5.2 – Caractéristiques des méthodes de groupement de valeurs de l'état de l'art comparativement à la méthode MODL.

		Critère							Algorithme			
		Arité X		Arité Y		Pénalisation						
	Méthode	Binaire	N-aire	Binaire	N-aire	Nombre de groupes	Sélection de modèles	Paramètres	Optimal	K-Moyennes	Echanges de valeurs	Ascendant
CHAID	[Kass, 1980]	X			X	X	X	X				X
Gini	[Breiman *et al.*, 1984]	X			X	X			X			
Twoing	[Breiman *et al.*, 1984]	X		X		X			X			
	[Chou, 1991]		X		X	X		X		X		
Gain ratio	[Quinlan, 1993]		X		X	X						X
	[Berckman, 1995]	X			X	X					X	
Tschuprow	[Ritschard *et al.*, 2001]		X		X							X
BIC	[Ritschard, 2003]		X		X		X					X
Khiops	[Boullé, 2004c]		X		X		X	X				X
MODL			X		X		X				X	X

des optimisations, puisqu'elle recherche une solution dans un espace de plus grande taille. Des méthodes de pré-optimisation et post-optimisation sont proposées afin de rechercher un compromis entre la qualité de la solution et la complexité algorithmique, super-linéaire comme dans le cas de la discrétisation.

L'originalité de la méthode par rapport aux méthodes alternatives les plus proches est l'utilisation d'une régularisation Bayesienne tenant compte explicitement de la nature catégorielle de la variable explicative, et la proposition d'heuristiques d'optimisation de complexité algorithmique super-linéaire.

Ces adaptations procurent à la méthode de groupement de valeurs les mêmes qualités que celles de la méthode de discrétisation MODL, comme l'indique le bilan (en section 4.2.5) de l'évaluation comparative des méthodes de prétraitement MODL.

5.3 Modèles en grille pour la régression

Cette section est issue des travaux de Carine Hue sur le positionnement de la méthode de discrétisation univariée MODL pour la régression, et est extraite pour l'essentiel de l'article [Hue and Boullé, 2007b].

Nous décrivons tout d'abord une situation particulière de l'apprentissage supervisé où

χ^2de Pearson	$X^2 = \sum_{i=1}^{I}\sum_{j=1}^{J} \frac{(NN_{ij} - N_{i.}N_{.j})^2}{NN_{i.}N_{.j}}$
Rapport de vraisemblance	$G^2 = 2\sum_{i=1}^{I}\sum_{j=1}^{J} N_{ij} \log \frac{NN_{ij}}{N_{i.}N_{.j}}$
Coefficient ϕ	$\phi = \sqrt{\frac{X^2}{N}}$
Contingence	$c_c = \sqrt{\frac{X^2}{N + X^2}}$
Tschuprow	$t = \sqrt{\frac{X^2}{N\sqrt{(I-1)(J-1)}}}$
Cramer	$v = \sqrt{\frac{X^2}{N \min((I-1)(J-1))}}$
τ de Goodman-Kruskal	$\tau_{y\leftarrow x} = \frac{N\sum_i \sum_j \frac{N_{ij}^2}{N_{i.}} - \sum_j N_{.j}^2}{N^2 - \sum_j N_{.j}^2}$
Coefficient d'incertitude u de Theil	$u_{y\leftarrow x} = \frac{N\log_2 N - \sum_i \sum_j N_{ij} \log_2 \frac{N_{i.}N_{.j}}{N_{ij}}}{N\log_2 N - \sum_j N_{.j}\log_2 N_{.j}}$

I	nombre de lignes
J	nombre de colonnes
N	nombre d'individus
$N_{i.}$	nombre d'individus de la ligne i
$N_{.j}$	nombre d'individus de la colonne j
N_{ij}	nombre d'individus de la case (i, j)

FIG. 5.1 – Critères d'association basés sur les effectifs d'un tableau de contingence , d'après [Ritschard *et al.*, 2001].

l'on s'intéresse à prédire le rang d'une variable numérique à expliquer plutôt que sa valeur. Nous exposons ensuite deux approches qui permettent de passer d'une prédiction ponctuelle en régression à une description plus fine de la loi prédictive. Nous positionnons alors notre contribution qui vise à fournir une estimation de la densité conditionnelle complète du rang d'une variable numérique par une approche Bayesienne non paramétrique.

5.3.1 Régression de valeur et régression de rang

En apprentissage supervisé on distingue généralement deux grands problèmes : la classification supervisée lorsque la variable à expliquer est catégorielle et la régression lorsqu'elle prend des valeurs numériques. Dans certains domaines tels que la recherche d'information, l'intérêt réside cependant plus dans le rang d'un individu par rapport à une variable plutôt que dans la valeur de cette variable. Par exemple, la problématique initiale des moteurs de recherche est de classer les pages associées à une requête et la valeur intrinsèque du score n'est qu'un outil pour produire ce classement. Indépendamment de la nature du problème à traiter, utiliser les rangs plutôt que les valeurs est une pratique classique pour rendre les modèles plus robustes aux valeurs atypiques et à l'hétéroscédasticité. En régression linéaire par exemple, un estimateur utilisant les rangs centrés dans l'équation des moindres carrés à minimiser est proposé dans [Hettmansperger and McKean, 1998]. L'apprentissage supervisé dédié aux variables ordinales est connu sous le terme de *régression ordinale* (cf. [Chu and Ghahramani, 2005] pour un état de l'art). Dans la communauté statistique les approches utilisent généralement le modèle linéaire généralisé et notamment le modèle cumulatif [McCullagh, 1980] qui fait l'hypothèse d'une relation d'ordre stochastique sur l'espace des variables explicatives.

En apprentissage automatique, plusieurs techniques employées en classification supervisée ou en régression métrique ont été appliquées à la régression ordinale : le principe de minimisation structurelle du risque dans [Herbrich *et al.*, 2000], un algorithme utilisant un perceptron appelé PRanking dans [Crammer and Singer, 2001] ou l'utilisation de machines à vecteurs de support dans [Shashua and Levin, 2002] [Chu and Keerthi, 2005]. Les problèmes considérés par ces auteurs comprennent cependant une échelle de rangs fixée au préalable et relativement restreinte (de l'ordre de 5 ou 10). Autrement dit, le problème se ramène à prédire dans quel partile se trouve la valeur à expliquer, en ayant défini les partiles avant le processus d'apprentissage. On se rapproche alors plus d'un problème de classification et les algorithmes sont évalués selon leur taux de bonne classification ou sur l'erreur de prédiction entre le vrai partile et le partile prédit.

5.3.2 Régression déterministe et régression probabiliste

Qu'il s'agisse de classification ou de régression, le modèle prédictif recherché est généralement ponctuel. On retient alors uniquement la valeur majoritaire en classification ou l'espérance conditionnelle en régression métrique. Ces indicateurs peuvent se révéler insuffisants, notamment pour prédire des intervalles de confiance mais également pour la prédiction de valeurs extrêmes. Dans ce contexte, la régression quantile ou l'estimation de densité permettent de décrire plus finement la loi prédictive.

La régression quantile vise à estimer plusieurs quantiles de la loi conditionnelle. Pour α réél dans $[0, 1]$, le quantile conditionnel $q_\alpha(x)$ est défini comme le réél le plus petit tel que la fonction de répartition conditionnelle soit supérieure à α. Reformulé comme la minimisation d'une fonction de coût adéquate, l'estimation des quantiles peut par exemple être obtenue par l'utilisation de splines [Koenker, 2005] ou de fonctions à noyaux [Takeuchi *et al.*, 2006]. Les travaux proposés dans [Chaudhuri *et al.*, 1994, Chaudhuri and Loh, 2002] combinent un partitionnement de l'espace des prédicteurs selon un arbre et une approche polynômiale locale. La technique récente des forêts aléatoires est étendue à l'estimation des quantiles conditionnels dans [Meinshausen, 2006]. En régression quantile, les quantiles que l'on souhaite estimer sont fixés à l'avance et les performances sont évaluées pour chaque quantile.

Les techniques d'estimation de densité visent à fournir un estimateur de la densité conditionnelle $P(Y|X)$. L'approche paramétrique présuppose l'appartenance de la loi conditionnelle à une famille de densités fixée à l'avance et ramène l'estimation de la loi à l'estimation des paramètres de la densité choisie. Les approches non paramétriques, qui s'affranchissent de cette hypothèse, utilisent généralement deux principes : d'une part, l'estimateur de la densité est obtenu en chaque point en utilisant les données contenues dans un *voisinage* autour de ce point ; d'autre part, une hypothèse est émise sur la forme recherchée localement pour cet estimateur. Très répandues, les méthodes dites à noyau définissent le voisinage de chaque point en convoluant la loi empirique des données par une densité à noyau centrée en ce point. La forme du noyau et la largeur de la fenêtre sont des paramètres à régler. Une fois la notion de voisinage définie, les techniques diffèrent selon la famille d'estimateurs visée : l'approche polynômiale locale [Fan *et al.*, 1996] regroupe les estimateurs constants, linéaires ou d'ordre supérieur. On peut également chercher à approximer la densité par une base de fonctions splines. Cette démarche d'estimation de la loi complète a déjà été adoptée en régression ordinale dans [Chu and Keerthi, 2005] en utilisant des processus Gaussiens dans un cadre Bayesien .

5.3.3 Positionnement de notre approche

Notre approche utilise la statistique d'ordre en amont du processus d'apprentissage. La manipulation exclusive des rangs au détriment des valeurs rend notre estimateur invariant par toute transformation monotone des données et peu sensible aux valeurs atypiques. Si le problème étudié ne s'intéresse pas à des variables ordinales mais à des variables numériques on peut bien entendu s'y ramener en calculant les rangs des individus à partir de leurs valeurs.

Contrairement aux problèmes habituellement traités en régression ordinale, on considère en amont de l'apprentissage l'échelle globale des rangs, de 1 au nombre d'individus dans la base. Notre méthode utilise l'information des variables explicatives pour mettre en évidence des plages de rangs de la variable à expliquer, dont le nombre n'est pas fixé à l'avance. La finesse de la description obtenue provient de l'approche Bayesienne utilisée pour la sélection de modèles, sans nécessiter aucun paramètre utilisateur.

Les modèles en grille pour la régression étendent de façon générique les méthodes MODL de prétraitement pour la classification supervisée, basées sur des modèles non

paramétriques pour l'estimation de la loi conditionnelle d'une variable à expliquer. L'extension au cas d'une variable à expliquer numérique se fait en prédisant les rangs de la variable à expliquer par plages, celles-ci étant déterminées automatiquement et optimalement grâce à l'approche Bayesienne utilisée pour la sélection de modèles.

5.4 Modèles en grille multivariés

Les modèles en grille multivariés sont des modèles génériques simples, basés sur des partitions élémentaires de chaque variable. Du fait de cette généricité, ils sont applicables à de nombreux problèmes de l'analyse des données, pour lesquels des approches comparables ont été étudiées. Dans cette section, nous présentons un éventail de ces méthodes pour les problèmes du regroupement des lignes et colonnes d'un tableau de contingence, et de la discrétisation multivariée étudiée dans plusieurs contextes de l'analyse de données.

5.4.1 Groupement des lignes et colonnes d'un tableau de contingence

Le problème du groupement des lignes et colonnes d'un tableau de contingence vise à constituer un tableau synthétique résumant l'information du tableau initial.

Dans [Ritschard *et al.*, 2001], l'objectif est de maximiser l'association entre les lignes et les colonnes d'un tableau de contingence. Une dizaine de mesures d'association classiques sont envisagées, comme par exemple les coefficients ϕ de Pearson, V de Cramer, T de Tschuprow, τ et γ de Goodman-Kruskal (cf. tableau 5.1). Ces mesures, symétriques ou asymétriques, s'appliquent selon les cas aux paires de variables numériques ou catégorielles. La taille du tableau de contingence final est trouvée automatiquement par maximisation du critère d'association. L'algorithme proposé est une heuristique gloutonne ascendante de fusion des lignes et des colonnes, similaire aux heuristiques utilisées classiquement dans les cas de la discrétisation et du groupement de valeurs. Le nombre d'évaluations de tableaux de contingence est quadratique ou cubique selon le type des variables, ce qui donne une complexité algorithmique en $O(N^4)$ dans le cas numérique et $O(N^5)$ dans le cas catégoriel. L'heuristique est évaluée dans [Ritschard, 2002] et la méthode est utilisée dans le cadre des arbres de décision [Zighed *et al.*, 2005] pour partitionner simultanément les variables explicatives et à expliquer, ce qui est utile quand la variable à expliquer contient de nombreuses valeurs.

Le problème du coclustering des lignes et des colonnes d'une matrice [Hartigan, 1972] est utilisé pour le groupement simultané des individus et des variables dans [Bock, 1979]. Dans une série d'articles [Govaert and Nadif, 2003, Govaert and Nadif, 2005, Nadif and Govaert, 2005, Govaert and Nadif, 2006], le problème du coclustering est abordé sous la forme d'un modèle de mélange par bloc, ce qui correspond à deux modèles de mélanges sur les individus et les variables s'exprimant l'un sur l'autre. L'algorithme EM (expectation maximisation) nécessite ici des extensions pour traiter efficacement les mélanges par bloc.

Dans [Dhillon *et al.*, 2003], le problème du coclustering de deux variables est évalué au moyen d'un critère issue de la théorie de l'information, en minimisant l'information

mutuelle entre les variables initiales et les variables groupées, pour une taille des partitions fixée par l'utilisateur. L'algorithme présenté consiste à affecter chaque ligne à son groupe de lignes, puis chaque colonne à son groupe de colonnes, en répétant le processus tant que l'amélioration du critère est non négligeable. Cet algorithme est analogue à l'algorithme des K-moyennes, en alternant l'optimisation des groupes de lignes et de colonnes. Des approches également basées sur des mesures de divergence informationnelle ont notamment été appliquées au bipartitionnement non supervisé des textes et des mots d'une base documentaire [Slonim and Tishby, 2000, El-Yaniv and Souroujon, 2001]. Une extension au cas de plus de deux variables est présentée dans [Bekkerman *et al.*, 2005], en réduisant l'interaction multivariée à l'ensemble des interactions entre paires de variables. L'algorithme d'optimisation proposé intègre des découpages aléatoires de clusters, des fusions entre clusters et des déplacements d'individus entre les clusters, pour une complexité algorithmique au moins cubique en le nombre d'individus.

5.4.2 Discrétisation multivariée

Les limites de la discrétisation univariée, aveugle aux interactions entre variables, ont fréquemment suscité des extensions au multivarié. Nous présentons dans cette section un échantillon des méthodes proposées dans la littérature, reflétant la diversité des approches et des domaines d'application.

Préparation des données. Dans [Muhlenbach and Rakotomalala, 2002], un graphe de voisinage est construit pour identifier des groupes d'individus ayant même valeur à expliquer, puis projeter ces groupes sur les variables explicatives pour obtenir les bornes d'une discrétisation multivariée. Dans [Chao and Li, 2005], chaque variable explicative est discrétisée en prenant en compte ses dépendances avec les autres variables au moyen d'une combinaison du critère d'entropie et du critère Relief [Kira and Rendell, 1992], utilisé classiquement en approche filtre de la sélection de variables.

Réseaux Bayesien. Dans le domaine de l'apprentissage des réseaux Bayesiens, la discrétisation multivariée est appliquée pour étendre l'approche usuelle reposant sur une discrétisation des variables préalable à l'apprentissage de la structure du réseau. Dans [Friedman and Goldszmidt, 1996], la structure du réseau et les discrétisations sont considérés alternativement, en apprenant une discrétisation multivariée localement à chaque noeud du réseau entre les variables connnexes. Le critère utilisé, basé sur une approche MDL [Rissanen, 1978], s'apparente à une extension au cas multivarié de la méthode univariée MDLPC [Fayyad and Irani, 1992]. L'approche proposée, validée sur des petites bases, pose des problèmes algorithmiques de tenue de charge selon les auteurs.

Dans [Monti and Cooper, 1999], un modèle de mélange paramétré par un nombre fini de composantes est utilisé en prétraitement pour identifier les corrélations entre variables. Cela permet d'obtenir une variable latente, correspondant aux clusters identifiés par le modèle de mélange. Cette variable latente est ensuite utilisée en tant que variable à expliquer pour discrétiser toutes les variables au moyen de méthodes de discrétisation supervisée classiques.

Règles de décision. Dans le domaine supervisé de l'apprentissage de règles de décision, [Kwedlo and Kretowski, 1999] utilisent un algorithme génétique pour apprendre un ensemble de règles de décision exploitant une discrétisation multivariée pour les variables explicatives numériques. Chaque chromosome encode une base de règles de décision multivariée, et représente une instance de modèle à optimiser. Le critère d'évaluation est établi de façon heuristique par un ratio entre le taux d'erreur du modèle et sa complexité, estimée en fonction du nombre de conditions utilisées dans la base de règles.

Règles d'association. Dans le domaine non supervisé des règles d'association, [Bay, 2001] décrit une méthode de discrétisation multivariée permettant de mettre en évidence les interactions entre variables numériques. L'algorithme est une heuristique gloutonne ascendante, initialisée par une discrétisation élémentaire non supervisée pour chaque variable. La finesse des discrétisations initiales élémentaires (en intervalles d'effectif ou de largeur égale) est un paramètre utilisateur, permettant essentiellement de contrôler le temps de calcul, important en raison de la complexité algorithmique en $O(N^K)$ (où K est le nombre de variables). Chaque fusion d'intervalles est évaluée au moyen d'un test de différence entre les distributions avant et après fusion de l'intervalle, basé sur des "contrast sets" [Bay and Pazzani, 1999].

Classification non supervisée. Dans le domaine de la classification non supervisée, la méthode CLIQUE [Agrawal *et al.*, 1998] recherche des clusters d'individus denses sur un sous-espace de l'espace initial ("sub-space clustering"). La méthode se base sur l'hypothèse que les groupes multivariés denses en dimension K restent denses après projection en dimension $K-1$. L'algorithme est initialisé en dimension 1 sur la base de discrétisations univariées en intervalles de largeur égale. Les clusters candidats en univarié sont alors identifiés, ce qui permet de sélectionner un nombre réduit de variables pertinentes. L'algorithme est ensuite appliqué sur les paires des variables pour identifier les clusters de variables en dimension 2, puis réitéré sur les dimensions supérieures. Le critère d'arrêt est basé sur la complexité des clusters, évaluée selon une approche inspirée du principe MDL [Rissanen, 1978].

La méthode MAFIA [Nagesh *et al.*, 2000] étend la méthode CLIQUE en supprimant la nécessité de paramètres utilisateurs et en recherchant des grilles adaptatives, par fusion d'intervalles adjacents. Une architecture de parallélisation des données et des traitements est également présentée, et évaluée sur des bases de plusieurs millions d'individus.

Complexité algorithmique de la discrétisation multivariée optimale. D'un point de vue algorithmique, le passage de la discrétisation univariée à la discrétisation multivariée est problématique. On passe ainsi d'un algorithme optimal en $O(N^3)$ dans le cas univarié à un problème NP-complet dans le cas multivarié. Dans le cas de la discrétisation supervisée bivariée, la recherche d'une bipartition consistante avec la variable à expliquer est étudiée dans [Chlebus and Nguyen, 1998]. Une bipartition est consistante si toutes ses régions sont pures, ce qui correspond à un taux d'erreur nul. Les auteurs montrent que le problème de la recherche d'une bipartition consistante pour un nombre de régions fixé est NP-complet. Dans [Elomaa *et al.*, 2005], le problème de minimisation du taux d'erreur

empirique sous contrainte de taille maximale des partitions est approximé au moyen d'une approche basée sur la programmation linéaire. La faisabilité en pratique de cette approche est encore à l'étude.

5.4.3 Positionnement de notre approche

Les modèles à base de décision discrète, très utilisés dans les méthodes d'apprentissage à partir des données, reposent sur une discrétisation des variables numériques. La discrétisation univariée étant par nature limitée, l'extension au multivarié a été étudié dans de nombreux contextes au moyen d'une grande variété d'approches.

Après avoir résumé de façon synthétique les approches décrites à notre connaissance dans la littérature, on présente les principaux apports des modèles en grille.

Synthèse de l'état de l'art

Les modèles de discrétisation ou groupements de valeurs multivariés sont utilisés dans de très nombreux contextes :

- modélisation synthétique des tableaux de contingence,
- prétraitements multivariés pour la classification supervisée,
- réseaux Bayesiens,
- règles de décision,
- règles d'association,
- recherche de clusters dans des sous-espaces pour la classification non supervisée,
- ...

Certaines méthodes combinent algorithmes et critères d'évaluation de façon intriquée, ce qui ne permet pas d'évaluer optimalité d'une solution. D'autres approches, plus nombreuses, explicitent d'une part un critère d'évaluation, d'autre part un algorithme d'optimisation.

Les critères d'évaluation proviennent d'une large diversité d'approches, plus encore que dans le cas univarié :

- critères basés sur le taux d'erreur,
- critères basés sur l'entropie,
- critères exploitant l'approche MDL,
- critères basés sur une mesure de similitude entre les données avant et après prétraitement de discrétisation multivariée,
- critères heuristiques, avec ou sans paramètres utilisateurs ou internes, dédiés à un domaine d'application,
- ...

Les algorithmes décrits dans la littérature témoignent de la complexité du problème. Les principales approches peuvent se résumer de la façon suivante :

- recherche de régions denses dans l'espace complet, puis projection de ces régions sur les variables pour obtenir des intervalles,
- recherche d'intervalles denses en univarié, puis reconstruction itérative de régions denses dans l'espace complet,
- optimisation directe d'un critère par une heuristique gloutonne ascendante.

Apports des modèles en grille

Les modèles en grille, analogues aux tables de décision [Kohavi, 1995], font partie des modèles les plus simples et les plus classiques de l'analyse des données. L'originalité de notre méthode provient non pas de la famille des modèles envisagée, mais de la démarche de modélisation, analysée en section 5.5.

Les apports des modèles en grille peuvent se résumer sur les axes suivants :
- unification de la modélisation des grilles, de l'univarié au multivarié, pour des variables numériques ou catégorielles, explicatives ou à expliquer,
- proposition d'un critère d'évaluation fondé théoriquement, ne procédant ni d'une approche heuristique, ni d'une approximation asymptotique,
- proposition d'algorithmes d'optimisation efficaces par rapport aux nombres N d'individus et K de variables, en $O(KN \log N \max(K, \log N))$ dans le cas numérique et $O(KN\sqrt{N} \log N \max(K, \log N))$ dans le cas le plus général.

Ces apports théoriques, évalués dans le chapitre 4, sont confirmés en pratique par une comparaison directe avec des méthodes alternatives dans le cas univarié. La comparaison est indirecte dans le cas de la classification supervisée où les classifieurs en grille sont confrontés à des méthodes alternatives lors de challenges internationaux. Des travaux importants sont nécessaires pour confirmer de façon comparative les apports des modèles en grille multivariée pour d'autres tâches de l'analyse de donnée.

5.5 Démarche de modélisation

Cette section analyse notre démarche de modélisation appliquée dans le cadre des modèles en grille, en la décrivant dans le cas le plus simple, celui de la méthode de discrétisation supervisée MODL. Après avoir rappelé les objectifs de modélisation, les choix de modélisation sont présentés et discutés.

5.5.1 Objectifs de la modélisation

Comme indiqué dans le chapitre 1, l'objectif général est l'automatisation de la phase de préparation des données dans le processus Data Mining. Plus précisément, il s'agit pour une tâche donnée de sélectionner un ensemble de variables pour construire un tableau individus * variables en entrée de la phase de modélisation.

Dans l'optique d'une automatisation, les critères de qualité retenus au chapitre 1 pour évaluer une méthode de sélection de variables en préparation des données sont :
- généricité,
- absence de paramétrage,
- fiabilité,
- finesse,
- interprétabilité,
- efficacité.

On se place ici dans le contexte d'une variable à expliquer catégorielle pour une unique variable explicative numérique. Sur la base d'un ensemble d'individus de taille finie disponible en apprentissage, il s'agit d'extraire automatiquement et efficacement un maximum

d'informations fiables, fines et interprétables, en minimisant les hypothèses sur les données.

5.5.2 Famille de modèles de discrétisation

On formalise les principaux choix de modélisation ayant conduit à l'espace des modèles considérés par la méthode MODL de discrétisation supervisée.

Choix 5.1. Modélisation de la probabilité conditionnelle.

Le choix d'une modélisation probabiliste plutôt que déterministe provient de l'objectif de généricité. En effet, seule la modélisation probabiliste peut s'adapter à toutes les applications potentielles de la préparation des données, que ce soit pour la classification, le scoring ou l'explication.

Choix 5.2. Modélisation non paramétrique.

L'objectif est de modéliser la probabilité de la variable à expliquer conditionnellement à la variable explicative. Comme on se place dans le contexte de la préparation des données pour le Data Mining, on ne dispose en général d'aucune connaissance du domaine. Le choix d'une méthode non paramétrique, ayant un comportement d'approximateur universel s'impose alors naturellement.

Précisons que l'on utilise le terme paramétrique au sens défini dans [Robert, 2006], c'est à dire quand les paramètres de modélisation appartiennent à un espace de dimension finie. La modélisation non paramétrique est intéressante essentiellement en raison de ses capacités d'approximation universelle, atteintes asymptotiquement. Peut-on pour autant considérer qu'un mélange infini de Gaussiennes, de polynômes de degrés quelconques ou de fonctions splines sont équivalents ? La question se pose notamment en pratique, puisque l'on ne dispose que d'échantillons de taille finie permettant d'estimer un nombre fini de paramètres. On privilégie dans le choix suivant une famille de modèles qui vise à maximiser l'expressivité d'un modèle quand le nombre de paramètres est borné.

Choix 5.3. Modèle de discrétisation n-aires.

Les modèles de discrétisation n-aires sont choisis pour leur expressivité, leur simplicité de mise en oeuvre et leur interprétabilité. Une discrétisation n-aire est ici considérée comme un modèle non paramétrique d'estimation de probabilité conditionnelle, constant par morceaux.

Ceci détermine la structure de l'espace $\mathcal{M}$ des paramètres des modèles, à savoir le choix du nombre I d'intervalles, des bornes b_i des intervalles, et pour chaque intervalle i des probabilités conditionnelles p_{ij} des J valeurs à expliquer. Pour un nombre d'intervalles I fixé, $\mathcal{M}_I$ est une famille paramétrique de modèles dont les paramètres $\{b_i\}_i$ sont à valeurs dans $\mathbb{R}^{I-1}$ et les paramètres $\{p_{ij}\}_{ij}$ à valeurs dans $[0,1]^{IJ}$. On peut maintenant préciser la nature non paramétrique de la famille de modèles envisagée $\mathcal{M} = \bigcup_{I \in \mathbb{N}^*} \mathcal{M}_I$, qui contient un nombre infini de paramètres, à valeurs dans les ensembles infinis non dénombrables $\mathbb{R}$ et $[0,1]$.

Choix 5.4. Modélisation sur les rangs.

La modélisation sur les rangs plutôt que sur les valeurs explicatives permet de se rendre indépendant de toute transformation monotone de la variable explicative et d'être plus robuste aux valeurs atypiques. On se conforme ainsi à l'objectif d'automatisation, en évitant les opérations usuelles de nettoyage, transformation, normalisation, fortes consommatrices de temps en préparation des données. De plus, l'invariance par transformation monotone de la variable explicative simplifie le problème de modélisation, sans perte d'expressivité.

La modélisation sur les rangs n'est pas propre à la méthode MODL. En effet, la quasi-totalité des méthodes de discrétisation de la bibliographie recherche les bornes des intervalles parmi les frontières entre les individus de l'échantillon d'apprentissage.

Choix 5.5. Modèle discret dépendant des données explicatives.

La famille de modèles de discrétisation dépend des individus en apprentissage pour le nombre d'intervalles maximum envisagé, pour le choix des bornes des intervalles, et localement à chaque intervalle, pour le choix des paramètres de la distribution à expliquer. Cela a pour conséquence de considérer une famille de modèles à paramètres discrets. Cette famille de modèles dépend des données explicatives uniquement, pas des données à expliquer, ce qui est acceptable puisque l'on modélise les probabilités conditionnelles uniquement, la distribution des valeurs explicatives étant exclue de l'objectif de modélisation. En contraignant les paramètres de modélisation par les données explicatives disponibles, on simplifie le problème de modélisation sans perte de généralité.

Dans le cas général, la modélisation dépendant des données est contestable en raison du double usage des données, qui augmente considérablement le risque de sur-apprentissage. Dans notre approche, on se limite à une dépendance vis-à-vis des données explicatives, qui elle évite le risque de sur-apprentissage. En effet, la famille des modèles est déterminée après avoir vu les données explicatives uniquement. Le meilleur modèle est ensuite choisi après avoir vu les données à expliquer.

De façon plus précise, soit un échantillon $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ de taille N, que l'on peut supposer sans perte de généralité indexé selon l'ordre des valeurs explicatives x_n. Suite aux choix précédents, on ne s'intéresse plus qu'à la modélisation de la séquence $D_Y = \{y_1, y_2, \ldots, y_N\}$ sous la forme d'une suite de I sous-séquences de taille N_i décrites par des distributions multinômiales de paramètres N_{ij}. Le passage de la famille $\mathcal{M}$ des modèles de discrétisation à la famille $\widehat{\mathcal{M}}$ des modèles de séquences ainsi définie est effectué en utilisant les données explicatives uniquement. Pour une taille d'échantillon N donnée, on se limite à la famille $\widehat{\mathcal{M}}^{(N)} = \bigcup_{I \in \{1,\ldots,N\}} \widehat{\mathcal{M}}_I^{(N)}$. Cette famille $\widehat{\mathcal{M}}^{(N)}$ ne dépend que de N et permet de décrire n'importe quelle séquence de N valeurs à expliquer. On peut considérer que $\widehat{\mathcal{M}} = \lim_{N \to \infty} \widehat{\mathcal{M}}^{(N)}$, avec pour chaque N une famille paramétrique $\widehat{\mathcal{M}}^{(N)}$ de modèles à nombre fini de paramètres de taille $\mathrm{O}(NJ)$, à valeurs dans l'ensemble $\{0, \ldots, N\}$ de taille finie. Par rapport à la famille initiale $\mathcal{M}$, l'espace des paramètres est considérablement réduit sans perte sur l'expressivité des modèles.

Modélisation continue et modélisation discrète.
L'hypothèse d'une variable explicative à valeurs dans $\mathbb{R}$ est très largement utilisée dans la littérature. En effet, quand la modélisation est effectuée au moyen de fonctions paramétriques ayant de bonnes propriétés analytiques (dérivabilité, intégrabilité...), cela permet

d'étudier formellement les propriétés asymptotiques des méthodes d'apprentissage statistique dans un cadre théorique maîtrisé.

Cette hypothèse de données à valeurs dans $\mathbb{R}$ est par contre peu réaliste en pratique, puisque les données à modéliser sont par nature discrètes et à support borné, au moins pour leur représentation informatique. L'utilisation d'espaces de fonctions à valeurs dans $\mathbb{R}$ correspond à l'expressivité du continu, infiniment riche en regard des données à modéliser, qui ne peuvent exprimer qu'un ensemble dénombrable de comportements. En se limitant au cas paramétrique usuel, l'expressivité est par contre très pauvre en l'absence de connaissances a priori sur le domaine. En définitive, la modélisation dans $\mathbb{R}$ est paradoxalement trop riche pour le domaine numérique considéré, et trop pauvre quand elle est limitée au cas paramétrique.

La modélisation discrète dépendant des données explicatives part du principe que tout paramètre de modélisation doit être introduit de façon parcimonieuse, uniquement s'il permet d'augmenter effectivement l'expressivité des modèles. Dans notre approche, la famille de modèles $\widehat{\mathcal{M}}$ n'exploite que la relation d'ordre entre les valeurs explicatives, en ne faisant aucune hypothèse sur le domaine des valeurs, qui n'est pas nécessairement $\mathbb{R}$, ni sur le type de distribution conditionnelle des valeurs à expliquer, qui peut être quelconque. Par rapport aux approches usuelles, on étend le domaine d'application de la méthode et on concentre l'expressivité des modèles sur les parties denses de l'espace des valeurs explicatives observables. L'utilisation d'espaces discrets permet de décrire un ensemble dénombrable de modèles de comportement, en adéquation avec l'ensemble dénombrable des comportements observables.

En abandonnant les hypothèses fortes sur les données habituellement supposées dans l'approche continue, on perd la possibilité d'exploitation des méthodes d'analyse numérique, généralement nécessaires pour l'obtention de preuves formelles de consistance, de convergence ou d'optimalité dans un cadre asymptotique. Notre approche discrète, qui n'effectue que très peu d'hypothèses sur les données, permet par contre d'exploiter des méthodes d'analyse combinatoire pour obtenir d'autres types de propriétés dans un cadre non asymptotique, comme celles démontrées en début de chapitre 4.

5.5.3 Approche Bayesienne de la sélection de modèles

La famille des paramètres de discrétisation étant explicitement définie, le choix de la meilleure discrétisation s'apparente alors à un problème de sélection de modèles. Le choix est fait de s'orienter vers un estimateur MAP selon une approche Bayesienne.

Dans le cadre Bayesien, la famille de modèles et sa distribution a priori sont définies au préalable. L'a priori sur les modèles permet d'exprimer des préférences avant d'avoir vu les données. De ce fait, notre approche ne se situe pas strictement dans ce cadre, puisque notre famille de modèles et notre a priori sur les modèles dépendent en partie des données. Comme la dépendance aux données ne concerne que la partie explicative et que les modèles ne s'intéressent qu'aux données à expliquer, nous adoptons néanmoins le cadre Bayesien. Cela permet de se focaliser sur une définition explicite de la distribution a priori des paramètres de discrétisation et sur le calcul des probabilités a priori et a posteriori, selon les principaux choix explicités ci-dessous.

Choix 5.6. Utilisation de la hiérarchie du paramétrage.

L'a priori utilisé exploite la hiérarchie naturelle du paramétrage, ce qui permet de limiter le nombre de fois où il faut exprimer des préférences sur le choix des paramètres. Par exemple, si l'on choisit une discrétisation en deux intervalles, une seule borne de discrétisation devra être choisie, au lieu de deux bornes pour une discrétisation en trois intervalles. En minimisant ainsi le nombre de choix à effectuer, la démarche s'apparente à l'approche MDL [Rissanen, 1978] qui vise la minimisation de la longueur de codage des paramètres des modèles. A chaque niveau de la hiérarchie des paramètres, le choix est uniforme, ce qui traduit notre ignorance sur les préférences entre les paramètres.

Choix 5.7. Indépendance entre les intervalles des distributions à expliquer.

Le choix est fait de supposer que les distributions des valeurs à expliquer sont indépendantes entre les intervalles. Cette option procure ici de nombreux avantages :
- l'interprétation des modèles est simplifiée, puisqu'elle se décompose sur les intervalles,
- la probabilité a priori de l'ensemble des distributions, ainsi que la vraisemblance conditionnelle des données, se factorise sur les intervalles,
- cela conduit à un critère additif, qui permet l'utilisation de méthodes d'optimisation de complexité algorithmique moindre.

Rappelons que ce choix d'indépendance par intervalles est plus faible que l'hypothèse usuelle IID d'indépendance entre les individus.

Choix 5.8. Consistance de la distribution a priori sur la suite des modèles $\widehat{\mathcal{M}}^{(N)}$.

Nous avons vu précédemment que notre famille de modèles non paramétriques $\widehat{\mathcal{M}}$ peut être considérée comme une famille limite pour les modèles paramétriques $\widehat{\mathcal{M}}^{(N)}$. Au lieu de définir explicitement une distribution a priori en dimension infinie sur $\widehat{\mathcal{M}}$, nous choisissons de spécifier la distribution a priori en dimension finie sur chaque famille $\widehat{\mathcal{M}}^{(N)}$, en se contraignant à respecter une propriété de consistance. Cette notion de consistance est intuitivement définie en stipulant que l'ordonnancement entre les modèles induit par leur probabilité a priori en dimension N doit être préservé en dimension $N+1$ quand ces modèles sont étendus par ajout d'un individu.

Illustrons notre démarche pour la définition de l'a priori sur un modèle de discrétisation pour N individus, défini par le nombre I d'intervalles, les effectifs N_i par intervalle et les effectifs N_{ij} par intervalle et par valeur à expliquer.

Pour le nombre d'intervalles, nous choisissons un a priori uniforme entre 1 et N. Cet a priori constant $P(I) = 1/N$ pour la famille $\widehat{\mathcal{M}}^{(N)}$, tend vers un a priori impropre quand N tend vers l'infini. Ce premier point montre comment notre démarche permet de contourner la difficulté à définir une distribution a priori dans le cas non paramétrique.

Pour le choix des effectifs des intervalles pour un nombre d'intervalles I donné, nous choisissons encore un a priori uniforme parmi toutes les partitions en I intervalles *potentiellement vides*, c'est à dire telles que $\sum_{i=1}^{I} N_i = N$ avec $N_i \geq 0$. On obtient un a priori constant $P(\{N_i\}_i) = 1/\binom{N+I-1}{I-1}$ consistant sur la suite des modèles $\widehat{\mathcal{M}}^{(N)}$. En effet, la probabilité a priori d'une discrétisation est décroissante avec le nombre d'intervalles I, quelle

que soit la taille N de l'échantillon. Notons que si l'on impose que les I intervalles soient *non vides*, ce qui peut sembler plus naturel, on aboutit à l'a priori $P(\{N_i\}_i) = 1/\binom{N-1}{I-1}$ qui lui ne respecte pas la propriété de consistance. En effet, une discrétisation en I intervalles est a priori préférée à une discrétisation en $I-1$ intervalles pour $N < 2I$, est cette préférence s'inverse pour $N > 2I$, comme illustré sur la figure 5.2.

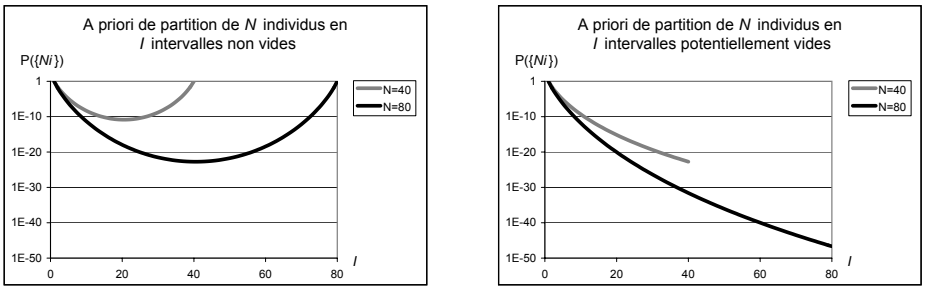


FIG. 5.2 – Comparaison de distributions a priori de partitions en intervalles. La distribution qui suppose les intervalles non vides n'est pas consistante, alors que la distribution qui admet les intervalles vides l'est : l'ordre sur le nombre d'intervalles est préservé quand la taille de l'échantillon augmente.

Cette notion de consistance permet intuitivement de limiter les risques liés à l'utilisation de modèles $\widehat{\mathcal{M}}^{(N)}$ dépendant des données. On se rapproche alors d'un cadre Bayesien strict, sans se limiter au cas paramétrique.

Remarque 5.14. *Des a priori alternatifs sont étudiés dans [Boullé, 2004b]. Par exemple, l'a priori uniforme sur l'ensemble de toutes les discrétisations s'apparente à une démarche de type maximum de vraisemblance, toutes les discrétisations étant a priori équiprobables. L'a priori uniforme conduit alors à un critère non pénalisé de type entropie, ce qui revient à sélectionner la discrétisation la plus "complexe" comportant autant d'intervalles que de valeurs. D'autres a priori sont également étudiés, intermédiaires entre l'a priori uniforme et l'a priori utilisant la hiérarchie sur les trois étages de paramétrage. Les propriétés analytiques des critères déduits de ces a priori alternatifs conduisent à les rejeter.*

Choix 5.9. Calcul des probabilités par des méthodes combinatoires.

Le calcul des probabilités a priori exploite la nature discrète de l'espace des paramètres des modèles, ce qui permet d'évaluer ces probabilités par simple comptage au moyen de méthodes combinatoires. Dans chaque intervalle, la distribution des valeurs à expliquer est décrite au moyen d'une loi multinômiale, et la vraisemblance conditionnelle au moyen d'un terme du multinôme. Comme pour les probabilités a priori, le calcul de la vraisemblance conditionnelle des données exploite ainsi le nombre fini d'individus dans l'échantillon et dans chaque intervalle, ce qui limite le nombre de choix alternatifs considérés à un nombre fini de possibilités. Cela permet d'obtenir un critère d'évaluation analytique pour le calcul de la probabilité a posteriori des paramètres de discrétisation.

Lien avec l'approche Bayesienne.
La pénalisation des discrétisations "complexes" provient ici de l'approche Bayesienne

utilisée : aucun paramétrage interne ou utilisateur n'est nécessaire pour exprimer des contraintes sur le nombre maximal d'intervalles ou sur l'effectif minimum par intervalle par exemple. L'utilisation de méthodes combinatoires pour obtenir un critère analytique exact permet un domaine d'application étendu, et notamment l'application de la méthode aux échantillons de très petite taille. En effet, aucune hypothèse de validité asymptotique n'est ici nécessaire.

Dans l'approche Bayesienne [Bernardo and Smith, 2000, Robert, 2006], deux choix sont critiques pour la qualité de la modélisation : le choix de la famille de modèles et le choix de la distribution a priori des modèles. Nous avons déjà traité le choix de la modélisation non paramétrique précédemment. En ce qui concerne le choix de la distribution a priori, on distingue le Bayesianisme subjectif [Goldstein, 2006] du Bayesianisme objectif [Berger, 2006]. Dans l'approche subjective, le choix de la loi a priori est le moyen de résumer l'information sur le phénomène à modéliser ainsi que l'incertitude sur ce phénomène. La détermination effective de la loi a priori est en pratique difficile, et dépend in fine de l'analyste de données. Dans l'approche objective, on recherche une distribution a priori minimalement informative et une procédure univoque permettant d'aboutir à cette distribution. Par exemple, [Jaynes, 2003] énonce le principe du maximum d'entropie qui consiste à définir une distribution a priori compatible avec les connaissances a priori et minimisant l'information apportée sur les paramètres des modèles. Dans le contexte de la préparation des données, nous nous sommes orientés vers une approche Bayesienne objective. Dans le cadre d'une modélisation paramétrique discrète, l'utilisation d'une loi a priori uniforme semble alors indiquée. Dans le cadre non paramétrique, cela conduit à une loi impropre sommant à l'infini, que l'on a cherché à éviter en s'orientant vers des méthodes issues de la théorie de l'information.

Lien avec la théorie de l'information.
Comme le log négatif d'une probabilité s'interprète comme une longueur de codage [Shannon, 1948], notre démarche de modélisation s'apparente à l'approche MDL [Rissanen, 1978, Hansen and Yu, 2001, Grünwald *et al.*, 2005], en cherchant à s'approcher de la complexité de Kolmogorov [Li and Vitanyi, 1997] pour le codage des données à expliquer. De même que la complexité de Kolmogorov est la longueur du plus petit programme informatique permettant de coder les données, notre a priori est guidé par une minimisation du nombre de choix à effectuer, et pour chaque choix du nombre d'alternatives à considérer. Le critère d'évaluation MODL d'une discrétisation s'apparente ainsi à une longueur de codage des données à expliquer.

Dans l'approche MODL, l'absence d'information prédictive se matérialise par la sélection du modèle nul de discrétisation, réduit à un seul intervalle. La longueur de codage de ce modèle nul est asymptotiquement équivalente à l'entropie des données à expliquer, ce qui correspond au codage des données telles quelles, sans l'intermédiaire d'un modèle prédictif. On est ici proche de la notion d'incompressibilité de Kolmogorov, qui définit le hasard par l'impossibilité de coder les données de façon plus concise que le codage des données telles quelles. Cela confère une grande robustesse à la méthode MODL, puisque la détection d'informations prédictives au moyen d'intervalles correspond nécessairement à un codage ayant un taux de compression meilleur que celui du modèle nul, donc à

l'identification de motifs ne provenant pas du hasard tel que défini par Kolmogorov.

La complexité de Kolmogorov est associée à la notion de *prior universel*, utilisable dans un cadre Bayesien pour la sélection de modèles [Vitányi and Li, 2000]. Intuitivement, les modèles les plus simples au sens de la complexité de Kolmogorov sont les plus probables. Cela permet de définir une approche MDL *idéale*, qui s'apparente à une approche Bayesienne objective pour laquelle la distribution a priori des modèles est choisie de façon univoque selon le prior universel. Malheureusement, la complexité de Kolmogorov n'est pas calculable, ce qui interdit l'utilisation en pratique du prior universel.

L'approche *practical MDL* aussi appelée *crude MDL* vise à se rapprocher du cadre idéal en codant les paramètres des modèles M et les données D connaissant les modèles en minimisant la longueur totale de codage $l(M) + l(D|M)$. Dans [Hansen and Yu, 2001], cette approche est illustrée dans le cas du codage d'une séquence de bits selon une loi de Bernoulli. Le codage obtenu est identique à celui de notre approche dans le cas d'un seul intervalle pour une variable à expliquer booléenne.

L'approche *practical MDL* n'est pas sans risque [Adriaans and Vitányi, 2006], puisqu'elle n'est valide qu'asymptotiquement dans le cas où la longueur de codage atteint la complexité de Kolmogorov, qui n'est pas calculable. Pour des codages moins efficaces, l'équilibre entre $l(M)$ et $l(D|M)$ (entre a priori et vraisemblance en termes Bayesiens) ne garantit pas la sélection du meilleur modèle. Notamment, dans le cadre non paramétrique, on peut trouver deux modèles M_1 et M_2 tels que $l(M_1)+l(D|M_1) \approx l(M_2)+l(D|M_2)$ avec $l(M_1) \ll l(M2)$. On risque alors de choisir un modèle très complexe en raison d'un choix de codage légèrement sous-optimal. C'est par exemple le cas pour un codage du choix des bornes des intervalles selon l'a priori "non consistant" $P(\{N_i\}_i) = 1/\binom{N-1}{I-1}$, utilisé dans [Dom, 1997] pour segmenter une séquence de bits en sous séquences distribuées selon des lois de Bernoulli. Avec ce codage, le modèle d'une séquence aléatoire de longueur N est légèrement plus probable a posteriori avec N intervalles qu'avec un seul intervalle, ce qui est un comportement que l'on cherche à éviter.

Dans notre démarche, nous avons suivi une approche de type *practical MDL*, en cherchant à coder le paramétrage du modèle de la façon la plus compacte possible. Ce codage est néanmoins recherché sous la contrainte de la consistance introduite dans le choix 5.8, ce qui garantit une cohérence des préférences a priori et aboutit à une sur-pénalisation de modèles dont la complexité $l(M)$ est proche de la complexité des données $l(D) \approx l(D|M_\emptyset)$. Cette propriété de consistance limite grandement les risques liés à l'application pratique du MDL, en favorisant la recherche des modèles autour des modèles de petite taille et en stabilisant les modèles sélectionnés pour différentes tailles d'échantillons d'apprentissage.

Lien avec la théorie de l'apprentissage statistique de Vapnik.
Dans le cas d'un problème de classification supervisée, Vapnik s'est intéressé au problème de sélection de modèles [Vapnik, 1996] en introduisant la VC-dimension (pour Vapnik-Chervonenkis dimension) notée $h(\mathcal{M})$ d'une famille de modèles $\mathcal{M}$. La VC-dimension, qui correspond au plus grand nombre d'individus séparables par un modèle de classification, permet de quantifier l'expressivité d'une famille de modèles. Vapnik a montré que le risque

(ou taux d'erreur) réel R^* est borné avec une probabilité η par le risque empirique R selon

$$R^* \leq R + \sqrt{\frac{h(\mathcal{M})(\log(2N/h(\mathcal{M})) + 1) - \log(\eta/4)}{N}}. \tag{5.1}$$

En adoptant une suite emboîtée de familles de modèles d'expressivité croissante, le principe SRM (structural risk minimization) permet de rechercher le meilleur modèle en minimisant la borne du risque réel.

La borne du risque réel est indépendante de la distribution des données, ce qui peut la rendre trop lâche pour un problème particulier. De plus, la VC-dimension d'une famille de modèles est souvent difficile à calculer, ce qui conduit à la majorer. En pratique, la borne du risque réel est souvent inutilisable pour la sélection de modèles. On utilise en général une procédure de validation croisée pour le choix des paramètres de modélisation, comme par exemple pour le choix du noyau et de la marge pour les classifieurs SVM (Support Vector Machine) [Vapnik, 1996].

Notre approche de la sélection de modèles se différencie de l'approche SRM de Vapnik sur différents aspects. Notre approche s'intéresse à la modélisation probabiliste dans le cas général alors que l'approche SRM se focalise sur le cas de la modélisation déterministe dans le cas de la classification supervisée. Notre approche est fortement dépendante des données et conduit en pratique à exploiter toutes les données disponibles pour sélectionner le meilleur modèle. L'approche SRM est indépendante de la distribution des données et fournit une borne en généralisation, permettant en théorie de sélectionner le meilleur modèle, mais nécessitant souvent en pratique un ensemble de validation.

5.5.4 Algorithmes d'optimisation

Le critère analytique de la méthode de discrétisation MODL est additif, dans le sens où il se décompose sur les intervalles. Cette propriété permet d'utiliser les méthodes d'optimisation existantes dans leur implémentation la plus performante.

Choix 5.10. Utilisation de l'heuristique gloutonne ascendante.

La méthode optimale, pertinente pour évaluer de façon comparative les méthode heuristiques, n'est pas utilisable en pratique en raison de sa complexité algorithmique en $O(N^3)$. Parmi les deux approches ascendante et descendante de l'heuristique gloutonne de discrétisation, l'approche ascendante est préférée puisqu'elle seule garantit une complexité algorithmique en $O(N \log N)$ dans le pire des cas.

Choix 5.11. Utilisation d'heuristiques de post-optimisation.

L'heuristique gloutonne ascendante est biaisée en faveur des discrétisations comportant trop d'intervalles, alors que l'heuristique gloutonne descendante l'est en faveur des discrétisations comportant trop peu d'intervalles. Contrairement aux préférences explicitées dans le choix de la distribution a priori des modèles, les biais algorithmiques sont implicites et difficiles à contrôler. Afin de réduire les biais algorithmiques, la méthode MODL incorpore des heuristiques de post-optimisation performantes, dans la limite de la complexité algorithmique $O(N \log N)$.

La méthode évalue systématiquement le modèle nul de discrétisation en un seul intervalle. Ainsi, quand la discrétisation sélectionnée comporte plusieurs intervalles, on a la garantie théorique que l'information identifiée n'est pas due au hasard au sens de Kolmogorov.

Remarque 5.15. *Plusieurs heuristiques d'optimisation et post-optimisation des discrétisations sont étudiées dans [Boullé, 2004b]. Cette étude confirme et précise les biais algorithmiques des heuristiques gloutonnes ascendantes et descendantes, qui permettent à des méthodes mal régularisées d'être robustes si elles sont optimisées de façon descendante, où qui au contraire conduisent des méthodes bien régularisées au sur-apprentissage si elles sont optimisées de façon ascendante.*

Remarque 5.16. *Les algorithmes de post-optimisation des discrétisations sont applicables que ce soit dans le cas des heuristiques gloutonnes ascendantes ou descendantes. Les évaluations effectuées dans [Boullé, 2004b] montrent qu'après post-optimisation, le biais algorithmique disparaît et les discrétisations obtenues sont statistiquement indiscernables de la discrétisation optimale. En éliminant ainsi le biais algorithmique, on obtient des discrétisations guidées uniquement par le biais explicite du critère, bénéficiant ainsi des avantages procurés par l'approche Bayesienne à l'origine de ce critère.*

5.5.5 Synthèse

Le problème de l'automatisation de la sélection de variables en préparation des données est un problème complexe, et toutes les informations disponibles sous la forme d'objectifs, de connaissances préalables, de données, d'hypothèses, de connaissance déduites, de techniques de sélection de modèles et d'optimisation sont utilisées pour le résoudre.

En résumé, notre démarche de modélisation se caractérise essentiellement par les éléments suivants :

- modèles non paramétriques d'estimation de densité,
- modèles dépendant des données explicatives,
- sélection de modèles empruntant à l'approche Bayesienne le choix explicite de la distribution a priori des modèles et la validité dans un cadre non asymptotique,
- choix de l'a priori guidé par le principe de longueur de description minimale, pour atteindre une bonne fiabilité dans un cadre asymptotique,
- algorithmes d'optimisation combinatoire poussés.

L'ensemble des évaluations menées au chapitre 4 procure une validation expérimentale claire de notre approche, qui peut être qualifiée de générique et interprétable en raison des modèles en grille envisagés, et de fine, fiable, sans paramètres et efficace en raison de la démarche de modélisation utilisée.

5.6 Conclusion

Les modèles en grille adressent le problème de la sélection de variables dans un cadre très générique, ce qui les rend utilisables dans de nombreux contextes de l'analyse de données. Nous avons choisi de positionner notre méthode par rapport aux méthodes alternatives les plus proches en préparation des données, puis par rapport aux techniques apparentées de sélection de modèles.

En préparation des données supervisée univariée, nous avons dressé une typologie des méthodes de discrétisation et groupement de valeurs de l'état de l'art. L'originalité principale de notre méthode provient essentiellement du choix de traiter le problème explicitement comme un problème de sélection de modèles, en intégrant les contraintes liées à l'automatisation de la préparation des données. Ainsi, notre méthode ne nécessite aucun paramètre utilisateur, et produit efficacement des estimations fines et fiables de la densité conditionnelle de la variable à expliquer, comme le rapporte l'évaluation du chapitre 4.

En régression, la grande majorité des méthodes existantes sont des méthodes de régression déterministe qui visent à prédire la valeur moyenne de la variable à expliquer. Notre méthode se situe dans une problématique qui n'est pas traitée à notre connaissance dans la bibliographie, celle de la régression probabiliste de rangs, et non de valeurs.

Quand on considère les modèles en grille de données dans le cas le plus général, on s'aperçoit que des modèles similaires ont été étudiés de nombreuses fois, au travers par exemple du regroupement des lignes et colonnes d'un tableau de contingence, de la discrétisation multivariée, des modèles à base de règles de décision ou de règles d'association, de la recherche d'hyper-rectangles denses dans des sous-espaces pour la classification non supervisée. Notre méthode se différencie par son approche générique, par son critère d'évaluation résultant d'une démarche explicite de sélection de modèles, et par ses algorithmes d'optimisation performants.

Un autre aspect important de notre approche est celui de la démarche de modélisation mise en oeuvre. Les modèles en grille sont des estimateurs non paramétriques de densité, dont les paramètres sont évalués avec un objectif pragmatique de validité dans un cadre non asymptotique. La démarche utilisée emprunte à la fois aux approches Bayesienne et MDL de la sélection de modèles, et ce qui plus atypique, se base sur des familles de modèles et d'a priori dépendant des données explicatives.

6

Bilan et perspectives

Sommaire

Comme rappelé en introduction, le sujet original des travaux décrits dans ce mémoire était :

> "Recherche d'une représentation des données efficace pour le Data Mining supervisé dans le cadre des grandes bases de données."

L'idée était d'explorer des espaces de représentation en étant guidé par la connaissance du domaine, puis de sélectionner le meilleur espace de représentation pour une tâche de classification supervisée. Il est vite apparu que pour atteindre cet objectif ambitieux, une première étape critique est de disposer d'une méthode de sélection de variables performante dans le contexte de la préparation des données. La section 6.1 dresse le bilan des modèles en grille de données introduits dans ce mémoire pour atteindre cette première étape. La section 6.2 présente l'impact des modèles en grille sur la phase de modélisation. La section 6.3 présente des extensions de notre approche, et propose quelques pistes pour atteindre l'objectif initial de ce mémoire.

6.1 Modèles en grille pour la sélection de variables

Modèles en grille. Les modèles en grille introduits dans ce mémoire constituent une famille de modèles non paramétriques pour l'estimation de densité. Chaque variable étant partitionnée en intervalles ou groupes de valeurs selon sa nature numérique ou catégorielle, l'espace complet des données est partitionné en une grille de cellules résultant du produit cartésien de ces partitions univariées. On recherche alors un modèle où l'estimation de densité est constante sur chaque cellule de la grille.

Du fait de leur très grande expressivité, les modèles en grille constituent des approximateurs de densité universels, difficiles à régulariser et à optimiser. Nous avons exploité une technique de sélection de modèles selon une approche Bayesienne, avec la spécification d'une distribution a priori des modèles en grille guidée par le principe de description de longueur minimale. L'originalité de notre démarche de modélisation est qu'elle est dépendante des données explicatives. De façon plus précise, la spécification précise des paramètres de modélisation et de leur distribution a priori est effectuée sur la base des données explicatives disponibles, et non de façon préalable au processus de sélection de modèles. Cette utilisation des données permet de simplifier grandement le problème en le transformant en un problème paramétrique à valeurs discrètes. On aboutit alors à une évaluation analytique de la probabilité a posteriori des modèles, qui fournit un critère optimisable au moyen d'algorithmes combinatoires.

Nous avons introduit des algorithmes d'optimisation exploitant les propriétés de notre critère d'évaluation et la faible densité des données dans les espaces de grande dimension. Ces algorithmes ont une complexité algorithmique garantie, linéaire en nombre d'individus en mémoire et super-linéaire en temps.

Application à la sélection de variable. Les modèles en grille de données sont naturellement **interprétables**, puisqu'ils s'expriment sous forme de règles simples portant sur les variables de l'espace de représentation. Ils sont **génériques**, puisqu'ils s'appliquent quels que soit la nature des variables, numérique ou catégorielle, et leur rôle, explicatif ou à expliquer, sans faire d'hypothèse sur la distribution et la quantité des données ni sur le domaine applicatif. La démarche de modélisation permet d'aboutir à un critère d'évaluation ne nécessitant **aucun paramètre**. Elle permet une détection **fiable** des informations contenues dans les données, ce qui est confirmé lors d'évaluations expérimentales intensives. Les algorithmes d'optimisation utilisés pour rechercher la meilleure grille de données permettent une sélection des variables **efficace**. Conjuguée avec la précision des critères d'évaluation établis dans un cadre non asymptotique, la performance des algorithmes d'optimisation permet une détection **fine** des informations dans les données. Les modèles en grille répondent ainsi à tous les critères de qualité identifiés en introduction pour évaluer une méthode de sélection de variables en préparation des données explicatives.

Finalisation de la méthode. Nous avons évalué les modèles en grille dans de nombreux contextes de la préparation des données, où ils se sont avérés performants. L'évaluation étant partielle, doit être complétée. Dans le cas des très grands nombres variables, nous avons notamment identifié les limites des algorithmes d'optimisation, qui doivent être améliorés pour explorer plus efficacement l'espace des variables. Enfin, il serait intéressant d'approfondir la formalisation de notre démarche de modélisation, en étudiant notamment les liens entre l'approche Bayesienne dans un cadre non paramétrique, l'approche MDL dans un cadre non asymptotique et les contraintes et apports des modèles dépendant des données explicatives.

6.2 Modèles en grille pour la modélisation

Classifieur Bayesien naïf et classifieur en grille. Nous avons utilisé les modèles en grille en tant que technique univariée de préparation des données pour le classifieur Bayesien naïf. Nous avons également exploité les modèles en grille directement en tant que technique de modélisation, en considérant un modèle en grille comme une table de décision multivariée. Alors que le classifieur Bayesien naïf évalue la densité conditionnelle multivariée en la factorisant sur les variables, le classifieur en grille détecte directement les interactions entre les variables. Les deux approches s'avèrent compétitives dans des domaines différents, et il serait pertinent d'envisager un classifieur hybride, évaluant la densité conditionnelle en la factorisant sur des groupes de variables modélisés par des grilles de données multivariées.

Moyennage de modèles. Nous avons proposé une nouvelle technique de moyennage de modèles basé sur le taux de compression des modèles, ce qui équivaut à un moyennage Bayesien de modèles avec lissage logarithmique de la distribution a posteriori des modèles. Nous avons évalué cette technique de moyennage avec deux famille de classifieurs, Bayesien naïf et en grille. Dans les deux cas, le moyennage s'est avéré très performant, en permettant à notre méthode de terminer en première position sur certains jeux de données lors de challenges internationaux. Dans le cas des modèles en grille, des investigations complémentaires sont nécessaires pour adapter les méthodes de moyennage.

Autres modèles prédictifs. Nous avons évalué les modèles en grille essentiellement dans le cas de la classification supervisée. Dans le cas de la régression et de la classification non supervisée, les résultats d'évaluation préliminaire sont prometteurs. Un travail d'évaluation comparative et d'exploration d'autres contextes de l'analyse des données est nécessaire pour exploiter tout le potentiel des modèles en grille.

6.3 Modèles de partitionnement pour l'évaluation de représentation

On propose dans cette section un cadre très général de la modélisation statistique et on montre que les modèle en grilles correspondent au cas de la représentation usuelle par tableau croisé individus * variables. On présente alors des pistes de généralisation de notre démarche de modélisation pour s'adresser à tous types de représentations.

Individus, représentations et modélisation statistique. On se donne un ensemble $\mathcal{O}$ d'individus observables et un ensemble $\mathcal{R}$ de représentations accessibles. Une représentation $R \in \mathcal{R}$ est une fonction $R : \mathcal{O} \to \mathcal{V}_R$, qui à tout individu associe une valeur d'un espace $\mathcal{V}_R$. Les représentations les plus usuelles sont les suivantes :

- R est une variable catégorielle quand $\mathcal{V}_R$ est un ensemble de taille finie,
- R est une variable ordonnée quand $\mathcal{V}_R$ est muni d'une relation d'ordre,
- R est une variable numérique quand $\mathcal{V}_R = \mathbb{R}$,

- R est une variable vectorielle quand $\mathcal{V}_R = \mathbb{R}^K$ pour une dimension K connue,
- R est une représentation tabulaire quand $\mathcal{V}_R = \mathcal{V}_{R_1} \times \mathcal{V}_{R_2} \times \ldots \times \mathcal{V}_{R_K}$ s'écrit comme un produit cartésien de représentations élémentaires ordonnées ou catégorielles pour une dimension K connue.

On dispose généralement de deux représentations particulières fixées : une représentation explicative R_1 à valeurs dans $\mathcal{V}_{R_1}$ et une représentation à expliquer R_2 à valeurs dans $\mathcal{V}_{R_2}$. L'objectif de la modélisation statistique dans son cadre le plus général consiste à décrire la représentation à expliquer des individus conditionnellement à leur représentation explicative, sur la base d'un échantillon d'individus de taille finie N extrait de $\mathcal{O}$.

La nature du problème de modélisation dépend des connaissances a priori dont on dispose sur les ensembles de valeurs $\mathcal{V}_{R_1}$ et $\mathcal{V}_{R_2}$. Par exemple, on parle de :

- discrétisation supervisée quand R_1 est une variable numérique et R_2 une variable catégorielle,
- groupement de valeur supervisée quand R_1 est une variable numérique et R_2 une variable catégorielle,
- classification supervisée quand R_1 est une représentation tabulaire et R_2 une variable catégorielle,
- régression quand R_1 est une représentation tabulaire (en général vectorielle) et R_2 une variable numérique,
- clustering quand R_1 est vide et R_2 est une représentation tabulaire (en général vectorielle).

Cette typologie de techniques de modélisation n'adresse qu'un petit sous-ensemble des problèmes potentiels, et en pratique on est conduit soit à se ramener à une représentation usuelle en transformant la représentation initiale, soit à étendre les techniques de modélisations usuelles aux caractéristiques de la représentation initiale.

Les modèles en grille revisités. Les modèles en grilles s'adressent de façon générique à tous les problèmes de modélisation quand les deux représentations explicatives et à expliquer sont des représentations tabulaires. Le spectre d'application est ainsi très large, puisqu'il couvre entre autres tous les problèmes usuels de classification supervisée, de régression et de clustering pour les représentations tabulaires sous la forme d'un tableau croisé individus * variables.

Un modèle en grille est un estimateur non paramétrique de densité reposant sur trois éléments : la spécification d'une partition $\mathcal{P}_1$ de l'ensemble $\mathcal{V}_1$ des valeurs explicatives, la spécification d'une partition $\mathcal{P}_2$ de l'ensemble $\mathcal{V}_2$ des valeurs à expliquer, et la spécification d'un modèle de distribution $\mathcal{D}_{1\to 2}$ des parties à expliquer pour chaque partie explicative.

La spécification des partitions exploite au maximum la connaissance a priori sur les ensembles $\mathcal{V}_1$ et $\mathcal{V}_2$ et la connaissance apportée par les individus de l'échantillon d'apprentissage. En univarié, les partitions sont constituées d'intervalles dont les bornes découlent des valeurs observés dans le cas numérique, de groupes de valeurs pour les valeurs observées dans le cas catégoriel. En multivarié, les partitions sont constituées du produit cartésien des partitions univariées.

La spécification pour chaque partie explicative de la distribution des parties à expliquer est effectuée sur la base d'une modélisation discrète dépendant de la taille de l'échantillon,

permettant une expressivité "exhaustive", puisque toutes les affectations des individus aux deux partitions sont considérées. Le choix du meilleur modèle est effectué au moyen d'une approche Bayesienne objective avec une distribution a priori des modèles en grille inspirée du prior universel de la théorie de l'information.

Un modèle en grille est donc un triplet $(\mathcal{P}_1, \mathcal{P}_2, \mathcal{D}_{1\to 2})$. La spécification des partitions $\mathcal{P}_1$ et $\mathcal{P}_2$ dépend des représentations alors que la spécification de la distribution $\mathcal{D}_{1\to 2}$ est générique. Il est alors naturel d'étendre les modèles en grilles à d'autres familles de partitions ou de représentations des données.

Extension à d'autres partitions des données. Dans le cas des variables catégorielles ou numériques, on peut envisager des partitions alternatives à celles étudiées dans le chapitre 2.

Dans le cas d'une variable catégorielle ayant de très nombreuses valeurs, une variante du modèle de groupement de valeurs consiste à définir un groupe spécial déterminé par un paramètre d'effectif minimal, pour accueillir les valeurs de faible effectif. Ce modèle étendu de groupement de valeurs est étudié dans [Boullé, 2005b].

Dans le cas d'une variable numérique, une variante du modèle de discrétisation consiste à imposer que les bornes des intervalles respectent une contrainte d'effectif égal par intervalle, ou de largeur égale par intervalle, comme dans le cas des histogrammes. La discrétisation supervisée optimale suivant ces contraintes est étudiée dans [Boullé, 2005c].

Extension à d'autres types de représentation. Toute information supplémentaire disponible sur l'ensemble des valeurs $\mathcal{V}_R$ d'une représentation peut être mise à profit pour améliorer la spécification d'un modèle de partition et de sa distribution a priori.

Par exemple, la méthode de discrétisation MODL d'une variable numérique R n'exploite que la relation d'ordre entre les valeurs de $\mathcal{V}_R$ pour spécifier une partition en intervalles. Si l'on ajoute une connaissance a priori sur le cardinal de $\mathcal{V}_R$, fini et connu, on peut tirer parti de cette information pour spécifier plus efficacement les distributions a priori et a posteriori des modèles de discrétisation. Ce type de représentation est étudié dans l'article [Boullé and Larrue, 2007] reproduit en annexe E dans le cas d'une image couleur, considérée comme une base de données de pixels pour laquelle on cherche à modéliser la densité des variables de couleur conditionnellement aux variables de position.

On peut également étendre l'approche au cas où l'on ne dispose pas de représentation directe des individus mais d'une mesure de similitude entre individus $S : \mathcal{O} \times \mathcal{O} \to \mathcal{V}_S$ à valeurs dans un ensemble $\mathcal{V}_S$ muni d'une relation d'ordre. Cette information est suffisante pour spécifier une partition des individus sur la base d'un ensemble d'individus prototypes, comme dans la méthode des K-médoïdes. Cette famille de partitions basée sur les mesures de similitude est étudiée dans [Ferrandiz, 2006, Ferrandiz and Boullé, 2006].

Automatisation de la recherche de représentation. En général on suppose que les individus sont tirés aléatoirement d'un ensemble $\mathcal{O}$, mais que les représentations explicatives R_1 et à expliquer R_2 sont fixées. En pratique, on dispose souvent d'un ensemble $\mathcal{R}$ de représentations possibles, de taille potentiellement infinie. La modélisation statistique est artificiellement divisée en deux tâches : la préparation des données vise à spécifier

une représentation explicative ou à expliquer, et la modélisation vise à décrire la représentation à expliquer des individus conditionnellement à leur représentation explicative. Il serait intéressant de pouvoir explorer l'ensemble $\mathcal{R}$ des représentations possibles et de prendre en compte l'incertitude sur le choix d'une représentation au moyen de méthodes statistiques.

Toute information préalable du domaine sur l'ensemble $\mathcal{R}$ des représentations possibles, sa structure, ou la façon dont les représentations sont construites peut être utilisé pour spécifier des modèles de partition des données, des a priori dépendant des données et appliquer la démarche de modélisation MODL.

Dans le cas des séries temporelles par exemple, on dispose en général pour chaque individu d'une série de mesures de taille potentiellement variable, que l'on cherche à résumer sous forme d'une représentation vectorielle. Les temps de mesure sont utilisables par exemple pour sélectionner un sous ensemble de variables équidistantes en temps, ou pour construire une représentation simplifiée par moyennage des mesures sur des intervalles de temps de taille fixe. De telles sélections ou constructions de variables sont également envisageables dans le cas du data mining spatial, où chaque variable est associée à des coordonnées. L'approche MODL peut alors tirer parti des ces informations pour sélectionner ou construire le meilleur sous-ensemble de variables en probabilisant la façon dont la représentation est construite, ce qui permet de rechercher la meilleure représentation dans un cadre statistique.

Si l'on dispose d'une spécification formelle de la structure des données sous une forme multi-tables par exemple, et d'une spécification formelle d'un langage de construction de variables, on peut envisager de repousser encore les limites de l'automatisation de la phase de préparation des données. Il s'agit une fois encore d'exploiter ces connaissances préalables sur les représentations pour spécifier des modèles de partitionnement des données, définir des a priori de sélection et construction de variable, et concevoir des algorithmes d'optimisation performants pour la recherche de la meilleure représentation.

Conclusion générale

Dans ce mémoire, nous avons introduit une famille de modèles en grille et une méthode de sélection de modèles permettant de rechercher une représentation des données efficace en préparation des données. Les modèles en grilles s'appliquent à tous les problèmes de modélisation statistique travaillant sur des représentations en tableaux croisés individus * variables. Notre démarche de modélisation peut être étendue à d'autre types de représentation des données et intégrer les informations du domaine applicatif dans un cadre statistique.

Annexes

A

MODL : a Bayes Optimal Discretization Method for Continuous Attributes

Mach Learn
DOI 10.1007/s10994-006-8364-x

MODL: A Bayes optimal discretization method for continuous attributes

Marc Boullé

Received: 5 April 2004 / Revised: 17 June 2005 / Accepted: 2 March 2006 / Published online: 8 May 2006

Abstract While real data often comes in mixed format, discrete and continuous, many supervised induction algorithms require discrete data. Efficient discretization of continuous attributes is an important problem that has effects on speed, accuracy and understandability of the induction models. In this paper, we propose a new discretization method MODL [1], founded on a Bayesian approach. We introduce a space of discretization models and a prior distribution defined on this model space. This results in the definition of a Bayes optimal evaluation criterion of discretizations. We then propose a new super-linear optimization algorithm that manages to find near-optimal discretizations. Extensive comparative experiments both on real and synthetic data demonstrate the high inductive performances obtained by the new discretization method.

Keywords Data mining · Machine learning · Discretization · Bayesianism · Data analysis

1. Introduction

Discretization of continuous attributes is a problem that has been studied extensively in the past (Catlett, 1991; Holte, 1993; Dougherty, Kohavi, & Sahami, 1995; Zighed & Rakotomalala, 2000; Liu et al., 2002). Many classification algorithms rely on discrete data and need to discretize continuous attributes, i.e. to slice their domain into a finite number of intervals. Decision tree algorithms first discretize the continuous attributes before they proceed with the attribute selection process. Rule-set learning algorithms exploit discretization methods to produce short and understandable rules. Bayesian network methods need discrete values to compute conditional probability tables.

Editor: Tom Fawcett

M. Boullé (✉)
France Telecom R&D, 2, Avenue Pierre Marzin, 22300 Lannion, France
e-mail: marc.boulle@francetelecom.com

[1] French patent No. 04 00179.

In the discretization problem, a compromise must be found between information quality (homogeneous intervals in regard to the attribute to predict) and statistical quality (sufficient sample size in every interval to ensure generalization). The chi-square-based criteria (Kass, 1980; Bertier & Bouroche, 1981; Kerber, 1991) focus on the statistical point of view whereas the entropy-based criteria (Catlett, 1991; Quinlan, 1993) focus on the information theoretical point of view. Other criteria such as Gini (Breiman et al., 1984) or Fusinter criterion (Zighed, Rabaseda, & Rakotomalala, 1998) try to find a trade off between information and statistical properties. The Minimum Description Length (MDL) criterion (Fayyad & Irani, 1992) is an original approach that attempts to minimize the total quantity of information both contained in the model and in the exceptions to the model. While most discretization methods are univariate and consider only a single attribute at a time, some multivariate discretization methods have also been proposed (Bay, 2001).

In this paper, we focus on univariate supervised discretization methods and propose a new method called MODL based on a Bayesian approach. First, we define a space of discretization models. The parameters of a specific discretization are the number of intervals, the bounds of the intervals and the class frequencies in each interval. Then, we define a prior distribution on this model space. Finally, we derive an evaluation criterion of discretizations, which is a direct application of the Bayesian approach for the discretization model space and its prior distribution. This criterion is minimal for the Bayes optimal discretization. Another important characteristic of the MODL discretization method is the search algorithm used to find optimal discretizations, i.e. discretizations which minimize the evaluation criterion. We describe an optimal search algorithm time complexity $O(n^3)$ where n is the sample size. We also propose a greedy search heuristic with super-linear time complexity and a new post-optimization algorithm that allows obtaining optimal discretizations in most cases. We demonstrate through numerous experiments that the theoretical potential of the MODL method leads to high quality discretizations.

The remainder of the paper is organized as follows. Section 2 presents the MODL method and its optimal evaluation criterion. Section 3 focuses on the MODL discretization algorithm. Section 4 proceeds with an extensive experimental evaluation both on real and synthetic data. Section 5 studies the relative contribution of the optimization criterion versus the search strategy.

2. The MODL evaluation criterion

The discretization methods have to solve a problem of model selection, where the data to fit is a string of class values and the model is a discretization model. The Bayesian approach and the MDL approach (Rissanen, 1978) are two techniques to solve this problem. In this section, we first recall the principles of these model selection techniques, and second present the MODL method, based on a Bayesian approach of the discretization problem.

2.1. Bayesian versus MDL model selection techniques

In the Bayesian approach, the best model is found by maximizing the probability $P(Model/\text{Data})$ of the model given the data. Using Bayes rule and since the probability $P(Data)$ is constant while varying the model, this is equivalent to maximizing:

$$P(Model)P(Data/Model). \quad (1)$$

Once the prior distribution of the models is fixed, the Bayesian approach finds the optimal model of the data, provided that the calculation of the probabilities $P(Model)$ and $P(Data/Model)$ is feasible.

To introduce the MDL approach, we can reuse the Bayes rule, replacing the probabilities by their negative logarithms. These negative logarithms of probabilities can be interpreted as Shannon code lengths, so that the problem of model selection becomes a coding problem. In the MDL approach, the problem of model selection is to find the model that minimizes:

$$DescriptionLength(Model) + DescriptionLength(Data/Model). \quad (2)$$

The relationship between the Bayesian approach and the MDL approach has been examined by Vitanyi and Li (2000). The Kolmogorov complexity of an object is the length of the shortest program encoding an effective description of this object. It is asymptotically equal to the negative log of a probability distribution called the *universal distribution*. Using these notions, the MDL approach turns into *ideal MDL*: it selects the model that minimizes the sum of the Kolmogorov complexity of the model and of the data given the model. It is asymptotically equivalent to the Bayesian approach with a universal prior for the model. The theoretical foundations of MDL allow focusing on the coding problem: it is not necessary to exhibit the prior distribution of the models. Unfortunately, the Kolmogorov complexity is not computable and can only be approximated.

To summarize, the Bayesian approach allows selecting the optimal model relative to the data, once a prior distribution of the models is fixed. The MDL approach does not need to define an explicit prior to find the optimal model, but the optimal description length can only be approximated and the approach is valid asymptotically.

2.2. The MODL optimal evaluation criterion

The objective of the discretization process is to induce a list of intervals that split the numerical domain of a continuous explanatory attribute. The data sample consists of a set of instances described by pairs of values: the continuous explanatory value and the class value. If we sort the instances of the data sample according to the continuous values, we obtain a string S of class values. In Definition 1, we introduce a space of discretization models.

Definition 1. A *standard* discretization model is defined by the following properties:

1. the discretization model relies only on the order of the class values in the string S, without using the values of the explanatory attribute,
2. the discretization model splits the string S into a list of substrings (the intervals),
3. in each interval, the distribution of the class values is defined by the frequencies of the class values in this interval.

Such a discretization model is called a SDM model.

Notation

n: number of instances
J: number of classes
I: number of intervals
n_i: number of instances in the interval i
n_{ij} number of instances of class j in the interval i

A SDM model is defined by the parameter set $\{I, \{n_i\}_{1\leq i\leq I}, \{n_{ij}\}_{1\leq i\leq I, 1\leq j\leq J}\}$.

This definition is very general and most discretization methods rely on SDM models. They first sort the samples according to the attribute to discretize (Property 1) and try to define a list of intervals by partitioning the string of class values (Property 2). The evaluation criterion is always based on the frequencies of the class values (Property 3).

Once a model space is defined, we need to fix a prior distribution on this model space in order to apply the Bayesian approach. The prior Definition 2 uses a uniform distribution at each stage of the parameters hierarchy of the SDM models. We also introduce a strong hypothesis of independence of the distributions of the class values. This hypothesis is often assumed (at least implicitly) by many discretization methods that try to merge similar intervals and separate intervals with significantly different distributions of class values. This is the case for example with the ChiMerge discretization method (Kerber, 1991), which merges two adjacent intervals if their distributions of class values are statistically similar (using the chi-square test of independence).

Definition 2. The following distribution prior on SDM models is called the *three-stage prior*:

1. the number of intervals I is uniformly distributed between 1 and n,
2. for a given number of intervals I, every division of the string to discretize into I intervals is equiprobable,
3. for a given interval, every distribution of class values in the interval is equiprobable,
4. the distributions of the class values in each interval are independent from each other.

Owing to the definition of the model space and its prior distribution, the Bayes formula is applicable to exactly calculate the prior probabilities of the models and the probability of the data given a model. Theorem 1 introduces the MODL evaluation criterion.

Theorem 1. *A SDM model distributed according to the three-stage prior is Bayes optimal for a given set of instances to discretize if the value of the following criterion is minimal:*

$$\log(n) + \log\binom{n+I-1}{I-1} + \sum_{i=1}^{I} \log\binom{n_i+J-1}{J-1} + \sum_{i=1}^{I} \log(n_i!/n_{i,1}!n_{i,2}!...n_{i,J}!). \tag{3}$$

Proof: The prior probability of a discretization model M can be defined by the prior probability of the parameters of the model $\{I, \{n_i\}_{1\leq i\leq I}, \{n_{ij}\}_{1\leq i\leq I, 1\leq j\leq J}\}$.

Let us introduce some notations:

- $p(I)$: prior probability of the number of intervals I,
- $p(\{n_i\})$: prior probability of the parameters $\{n_1, \ldots, n_I\}$,
- $p(n_i)$: prior probability of the parameter n_i,
- $p(\{n_{ij}\})$: prior probability of the parameters $\{n_{11}, \ldots, n_{ij}, \ldots, n_{IJ}\}$,
- $p(\{n_{ij}\}_i)$: prior probability of the parameters $\{n_{i1}, \ldots, n_{iJ}\}$.

The objective is to find the discretization model M that maximizes the probability $p(M/S)$ for a given string S of class values. Using Bayes formula and since the probability $p(S)$ is constant under varying the model, this is equivalent to maximizing $p(M)\, p(S/M)$.

Let us first focus on the prior probability $p(M)$ of the model. We have

$$\begin{aligned} p(M) &= p(I, \{n_i\}, \{n_{ij}\}) \\ &= p(I)p(\{n_i\}/I)p(\{n_{ij}\}/I, \{n_i\}). \end{aligned}$$

The first hypothesis of the three-stage prior is that the number of intervals is uniformly distributed between 1 and n. Thus we get

$$p(I) = \frac{1}{n}.$$

The second hypothesis is that all the divisions of S into I intervals are equiprobable for a given I. Computing the probability of one set of intervals turns into the combinatorial evaluation of the number of possible interval sets. Dividing the string S into I intervals is equivalent to decomposing the natural number n as the sum of the frequencies n_i of the intervals. Using combinatorics, we can prove that the number of choices of such any $\{n_i\}_{1 \leq i \leq I}$ is equal to $\binom{n+I-1}{I-1}$. Thus we obtain

$$p(\{n_i\}/I) = \frac{1}{\binom{n+I-1}{I-1}}.$$

The last term to evaluate can be rewritten as a product using the hypothesis of independence of the distributions of the class values between the intervals. We have

$$\begin{aligned} p(\{n_{ij}\}/I, \{n_i\}) &= p(\{n_{ij}\}_1, \{n_{ij}\}_2, \ldots, \{n_{ij}\}_I/I, \{n_i\}) \\ &= \prod_{i=1}^{I} p(\{n_{ij}\}_i/I, \{n_i\}) \\ &= \prod_{i=1}^{I} p(\{n_{ij}\}_i/n_i). \end{aligned}$$

For a given interval i with size n_i, all the distributions of the class values are equiprobable. Computing the probability of one distribution is a combinatorial problem, which solution is:

$$p(\{n_{ij}\}/n_i) = \frac{1}{\binom{n_i+J-1}{J-1}}.$$

Thus,

$$p(\{n_{ij}\}/I, \{n_i\}) = \prod_{i=1}^{I} \frac{1}{\binom{n_i+J-1}{J-1}}.$$

The prior probability of the model is then

$$p(M) = \frac{1}{n} \frac{1}{\binom{n+I-1}{I-1}} \prod_{i=1}^{I} \frac{1}{\binom{n_i+J-1}{J-1}}.$$

Let us now evaluate the probability of getting the string S for a given model M. We first split the string S into I sub-strings S_i of size n_i and use again the independence assumption

between the intervals. We obtain

$$\begin{aligned} p(S/M) &= p(S/I, \{n_i\}, \{n_{ij}\}) \\ &= p(S_1, S_2, ..., S_I/I, \{n_i\}, \{n_{ij}\}) \\ &= \prod_{i=1}^{I} p(S_i/I, \{n_i\}, \{n_{ij}\}) \\ &= \prod_{i=1}^{I} \frac{1}{(n_i!/n_{i,1}!n_{i,2}!...n_{i,J}!)}, \end{aligned}$$

as evaluating the probability of a sub-string S_i under uniform prior turns out to be a multinomial problem.

Taking the negative log of the probabilities, the maximization problem turns into the minimization of the claimed criterion

$$\log(n) + \log\binom{n+I-1}{I-1} + \sum_{i=1}^{I} \log\binom{n_i+J-1}{J-1} + \sum_{i=1}^{I} \log(n_i!/n_{i,1}!n_{i,2}!...n_{i,J}!).$$

□

The first term of the criterion corresponds to the choice of the number of intervals and the second term to the choice of the bounds of the intervals. The third term represents the choice of the class distribution in each interval and the last term encodes the probability of the data given the model.

When the MODL criterion is used, we prove in Theorem 2 that optimal splits always fall on boundary points, where boundary points are located between two instances in the string S having different class values. The same property have already be presented for the MDLPC criterion in Fayyad and Irani (1992) and for a larger class of impurity measures in Elomaa and Rousu (1996).

Theorem 2. *In a Bayes optimal SDM model distributed according to the three-stage prior, there is no split between two instances related to the same class.*

Proof: Assume that, contrary to the claim, such a split exists between two instances related to the same class (indexed as class 1 for convenience reasons). Let A1 and B1 be the intervals from each side of the split.

We construct two new intervals A0 and B2 by moving the last instance from A1 to B1. The cost variation $\Delta Cost1$ of the discretization is

$$\begin{aligned} \Delta Cost1 = &\log((n_{A0}+J-1)!/n_{A0}!(J-1)!) + \log(n_{A0}!/n_{A0,1}!n_{A0,2}!...n_{A0,J}!) \\ &+ \log((n_{B2}+J-1)!/n_{B2}!(J-1)!) + \log(n_{B2}!/n_{B2,1}!n_{B2,2}!...n_{B2,J}!) \\ &- \log((n_{A1}+J-1)!/n_{A1}!(J-1)!) - \log(n_{A1}!/n_{A1,1}!n_{A1,2}!...n_{A1,J}!) \\ &- \log((n_{B1}+J-1)!/n_{B1}!(J-1)!) - \log(n_{B1}!/n_{B1,1}!n_{B1,2}!...n_{B1,J}!). \end{aligned}$$

The frequencies are the same for each class except for class 1, thus

$$\begin{aligned}\Delta Cost1 &= \log((n_{A1}+J-2)!/(n_{A1}+J-1)!) + \log((n_{B1}+J)!/(n_{B1}+J-1)!)\\ &\quad + \log(n_{A1,1}!/(n_{A1,1}-1)!) + \log(n_{B1,1}!/(n_{B1,1}+1)!)\\ &= \log((n_{B1}+J)/(n_{A1}+J-1)) + \log(n_{A1,1}/(n_{B1,1}+1))\\ &= \log((n_{B1}+1)/(n_{B1,1}+1)) - \log(n_{A1}/n_{A1,1})\\ &\quad + \log((n_{B1}+J)/(n_{B1}+1)) - \log((n_{A1}+J-1)/n_{A1}).\end{aligned}$$

Using the property $1 \leq x < y \Rightarrow (y+1)/(x+1) < y/x$, we get

$$\begin{aligned}\Delta Cost1 &< \log(n_{B1}/n_{B1,1}) - \log(n_{A1}/n_{A1,1})\\ &\quad + \log((n_{B1}+J)/(n_{B1}+1)) - \log((n_{A1}+J)/(n_{A1}+1)).\end{aligned}$$

Similarly, we construct two intervals A2 and B0 by moving the first instance from B1 to A1. This time, the cost variation $\Delta Cost2$ of the discretization subject to

$$\begin{aligned}\Delta Cost2 &< \log(n_{A1}/n_{A1,1}) - \log(n_{B1}/n_{B1,1})\\ &\quad + \log((n_{A1}+J)/(n_{A1}+1)) - \log((n_{B1}+J)/(n_{B1}+1)).\end{aligned}$$

We notice that the upper bounds of the two cost variations $\Delta Cost1$ and $\Delta Cost2$ have exactly opposite values. Therefore, the cost variation of the discretization is strictly negative and the initial discretization could not be optimal. As this is contradictory with the initial assumption, the claim follows. □

Based on Theorem 2, important reductions in time consumption can be obtained in the optimization algorithms, since only the boundary points need to be evaluated to find optimal or near-optimal discretizations.

Theorem 3. *In a Bayes optimal SDM model distributed according to the three-stage prior, there is no pair of adjacent intervals each containing one single instance.*

Proof: Let A and B be two adjacent intervals each containing one single instance related to different classes (otherwise, the intervals should be merged according to Theorem 2). We compute the cost variation $\Delta Cost$ of the discretization after the merge of the two intervals into a new interval $A \cup B$, bringing the number of intervals from I down to $I-1$.

$$\begin{aligned}\Delta Cost &= \log\binom{n+I-2}{I-2} - \log\binom{n+I-1}{I-1}\\ &\quad + \left(\log\binom{n_{A\cup B}+J-1}{J-1} + \log(n_{A\cup B}!/n_{A\cup B,1}!n_{A\cup B,2}!\ldots n_{A\cup B,J}!)\right)\\ &\quad - \left(\log\binom{n_A+J-1}{J-1} + \log(n_A!/n_{A,1}!n_{A,2}!\ldots n_{A,J}!)\right)\\ &\quad - \left(\log\binom{n_B+J-1}{J-1} + \log(n_B!/n_{B,1}!n_{B,2}!\ldots n_{B,J}!)\right)\end{aligned}$$

$$\Delta Cost = \log\,(I - 1/n + I - 1) + \log((n_{A\cup B} + J - 1)!\,(J-1)!/(n_A + J - 1)!(n_B + J - 1)!) - \sum_{j=1}^{J} \log \binom{n_{A\cup B,j}}{n_{A,j}}$$

Since $n_A = n_B = 1$ and $n_{A\cup B} = 2$, we obtain:

$$\Delta Cost = \log(I - 1/n + I - 1) + \log(J + 1/J).$$

$$\Delta Cost \leq 0 \Leftrightarrow I \leq nJ + 1.$$

The cost variation is always strictly negative after the merge of two adjacent singleton intervals. The claim follows. □

The main interest of Theorem 3 is to provide an intuitive evaluation of the asymptotic behaviour of the three-stage prior: a discretization model based on two adjacent singleton interval is not optimal for a good generalization. Theorem 4 is another interesting property resulting from the three-stage prior.

Theorem 4. *In a SDM model distributed according to the three-stage prior and in the case of two classes, the discretization composed of one single interval is more probable than the discretization composed of one interval per instance.*

Proof: Let $Cost_1$ and $Cost_n$ be the value of the MODL criterion in the case of one interval and of n intervals.

$$Cost_1 = \log\,(n) + \log\,(n+1) + \log(n!/n_1!n_2!).$$

$$Cost_n = \log(n) + \log((2n-1)!/(n-1)!n!) + n\log(2).$$

Since $n + 1 = (n+1)!/1!n! \leq (2n-1)!/(n-1)!n!$ and $n!/\,n_1!n_2! < 2^n$, the claim follows. □

A close inspection of formula 3 reveals that the second term, which encodes the number of choices when dividing the string S into I intervals, includes the possibility of empty intervals. This is a deliberate choice in the three-stage prior that favours discretizations with small numbers of intervals. If we exclude this possibility, the Theorems 2–4 are no longer true. These empty intervals do not need to be explored in optimization algorithms since adding empty intervals is always penalized by an increase of the evaluation criterion.

To summarize, the MODL discretization method is directly based on the Bayesian approach. The definitions of the SDM models and the three-stage prior are both general enough to capture the complexity of real data and simple enough to allow an exact calculation of the probabilities involved in the Bayes rule. This provides the guarantee of optimality in the choice of the discretization model, in the context of the three-stage prior. The bias resulting from the choice of this prior leads to interesting demonstrable properties.

Table 1 Dynamic programming algorithm

Let $S^{i,j}$ be the substring of S consisting of the instances from i to j. $S^{1,n} = S$.
Let Disc($S^{i,j}$, k) be the optimal discretization of $S^{i,j}$ into exactly k intervals.
–For each k, $1 \leq k \leq n$,
–For each j, $1 \leq j \leq n$,
–If k=1, Disc($S^{1,j}$, 1) = $\{S^{1,j}\}$
–If k>1, Disc($S^{1,j}$, k) is obtained by minimizing the MODL value of all discretizations Disc($S^{1,i}$, k−1) U $\{S^{i+1,j}\}$ for $1 \leq i \leq j$.

3. The MODL algorithm

Once the optimality of this evaluation criterion is established, the problem is to design a search algorithm in order to find a discretization model that minimizes the criterion. In this section, we present three algorithms that represent different trade-off between the time complexity of the search algorithm and the quality of the discretizations.

3.1. Optimal algorithm

The MODL evaluation criterion consists of an evaluation of the partition of the string S into I intervals (first and second term in formula 3) and of the sum of the evaluation of the intervals S_i (third and last term in formula 3). The first part of the MODL criterion (value of the partition) depends only upon the sample size n and the number of intervals I and the second part (value of the intervals) is cumulative on the intervals. Hence, if a partition of S into I intervals S_1, $S_2, \ldots S_I$ is a MODL optimal discretization of S, then a partition of $S-S_1$ into I–1 intervals $S_2, \ldots S_I$ is a MODL optimal discretization of $S-S_1$. This interesting property is sufficient to adapt the dynamic programming algorithm presented in Fischer (1958), Lechevallier (1990), Fulton, Kasif, and Salzberg (1995) and Elomaa and Rousu (1996). We summarize in Table 1 this dynamic programming algorithm applied to the MODL discretization method.

The main loop of the algorithm finds the optimal discretizations of S into exactly k intervals ($1 \leq k \leq n$), and thus allows to obtain the optimal MODL discretization of the string S. The algorithm runs in $O(n^3)$ time. Although it is not applicable in the case of large databases, this optimal algorithm is helpful to evaluate the quality of search heuristics such as these presented in the next sections.

3.2. Greedy heuristic

In this section, we present a standard greedy bottom-up heuristic. The method starts with initial single value intervals and then searches for the best merge between adjacent intervals. This merge is performed if the MODL value of the discretization decreases after the merge and the process is reiterated until not further merge can decrease the criterion.

With a straightforward implementation of the algorithm, the method runs in $O(n^3)$ time. However, the method can be optimized in $O(n \log(n))$ time owing to an algorithm similar to that presented in Boullé (2004) and summarized in Table 2. The algorithm is mainly based on the additivity of the evaluation criterion. Once a discretization is evaluated, the value of a new discretization resulting from the merge between two adjacent intervals can be evaluated in a single step, without scanning all the other intervals. Minimizing the value of the discretizations after the merges is the same as maximizing the related variation of value Δvalue. These Δvalues can be kept in memory and sorted in a maintained sorted list (such as

Table 2 Optimized greedy bottom-up merge algorithm

Initialization
–Sort the explanatory attribute values: O(n log(n))
–Create an elementary interval for each value: O(n)
–Compute the value of this initial discretization: O(n)
–Compute the Δvalues related to all the possible merges: O(n)
–Sort the possible merges: O(n log(n))
Optimization of the discretization
Repeat the following steps: at most n steps
–Search for the best possible merge: O(1)
–Merge and continue if the best merge decreases the discretization value
–Compute the Δvalues of the two intervals adjacent to the merge: O(1)
–Update the sorted list of merges: O(log(n))

an AVL binary search tree for example), or more simply in a priority-queue. After a merge is completed, the Δvalues need to be updated only for the new interval and its adjacent intervals to prepare the next merge step.

3.3. Post-optimization algorithm

Compared to the optimal algorithm, the greedy heuristic is time efficient, but it may fall into a local optimum. First, the greedy heuristic may stop too soon and produce too many intervals. Second, the boundaries of the intervals may be sub-optimal since the merge decisions of the greedy heuristic are never rejected. Given that the MODL criterion is optimal, a time efficient post-optimization of the discretization is meaningful.

We propose a new post-optimization algorithm based on hill-climbing search in the neighborhood of a discretization. The neighbors of a discretization are defined with combinations of interval splits and interval merges, as pictured in Fig. 1.

In a first stage called *exhaustive merge*, the greedy heuristic merge steps are performed unconditionally until the discretization consists of a single interval. The best encountered discretization is then memorized. This stage allows escaping local minima with several successive merges and needs O(n log(n)) time.

In a second stage called *greedy post-optimization*, all the neighbours of the best discretization consisting of Splits, MergeSplits and MergeMergeSplits are evaluated. The best improvement of the discretization is performed if the evaluation criterion decreases, and this steps is reiterated until no neighbour can decrease the value of the discretization. The calculation of all these discretization neighbours can be done in O(n) time and inserted in sorted

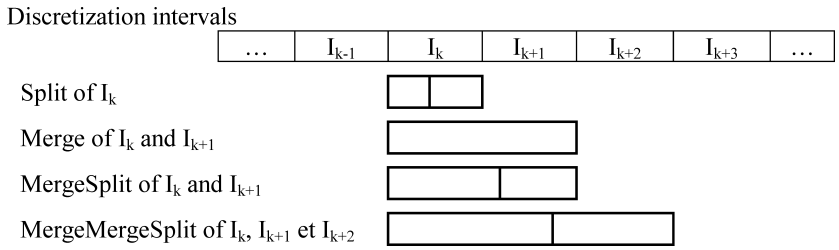


Fig. 1 Combinations of interval splits and interval merges used to explore the neighborhood of a discretization

lists in O(n log(n)) time, like in the greedy bottom-up merge algorithm. Each improvement of the discretization requires O(log(n)) time to maintain the data structures kept in memory. This second stage converges very quickly and requires only a few steps, so that its overall time complexity is still O(n log(n)).

The post-optimization holds two straightforward notable properties. The first one is that post-optimizing can only improve the results of the standard greedy heuristic since its comes after. The second one is that in case of attributes whose optimal discretization consists of one single interval, the optimum is necessarily found owing to the exhaustive merge stage of the post-optimization.

4. Experiments

In our experimental study, we compare the MODL discretization method with other discretization algorithms. In this section, we introduce the evaluation protocol, the alternative evaluated discretization methods and expose the evaluation results on real and artificial datasets.

4.1. The evaluation protocol

Discretization is a general purpose preprocessing method that can be used for data exploration or data preparation in data mining. While they are critical in the case of decision tree methods, discretization methods can also be used for bayesian networks, rule-set algorithms or logistic regression. However, discretization methods have mainly been evaluated using decision trees (Fayyad & Irani, 1992; Kohavi & Sahami, 1996; Elomaa & Rousu, 1999; Liu et al., 2002) and less frequently using naïve Bayes methods (Dougherty et al., 1995). In their experiments, Kohavi and Sahami (1996) report that entropy-based discretization methods perform better that error-based methods. Maximizing the accuracy on each attribute can hide the variations of the class conditional density and hinder improvements of the classifier accuracy when the attributes are combined. In the case of naïve Bayes classifiers, Dougherty et al. (1995) demonstrate that using any discretization algorithm outperforms the naïve Bayes algorithm with the normality assumption for continuous attributes. Elomaa and Rousu (1999) show that using optimal multi-splitting discretization algorithms does not bring a clear accuracy advantage over binary splitting in the case of decision trees.

Although these evaluations bring many insights on the impact of discretization methods on classifiers, the contribution of the discretization is not always clear. For example, decision tree are composed of several modules including a preprocessing algorithm, a selection criterion, a stopping rule and a pruning algorithm. The discretization preprocessing step can be done once at the root of the tree or repeated at each node of the tree. It can use a binary-splitting strategy or a multi-splitting strategy (better partition, but more fragmented subsequent data). The performances of such classifiers result from complex interactions between these modules and strategies.

In order to evaluate the intrinsic performance of the discretization methods and eliminate the bias of the choice of a specific induction algorithm, Zighed, Rabaseda, and Rakotomalala (1999) consider each discretization method as an elementary inductive method that predicts the local majority class in each learned interval. They apply this approach on the waveform dataset (Breiman, 1984) to compare several discretization methods on the accuracy criterion.

We extend this protocol as in Boullé (2003) and evaluate the elementary discretization inductive methods using all the continuous attributes contained in several datasets. This allows

us to perform hundreds of experiments instead of at most tens of experiments. The discretizations are evaluated for three criteria: accuracy, robustness (test accuracy/train accuracy) and number of intervals. To facilitate the interpretation of the very large set of experimental results, we first proceed with a multi-criteria analysis based on gross global summaries of the results. We then come to a more detailed analysis by the means of curves representing all the experiments sorted by increasing difference with the MODL method. Finally, we zoom into some specific experiments in order to provide both an illustration on some sample MODL discretizations and an explanation of typical differences of behavior between the MODL method and the alternative methods. In addition, we perform extensive experiments on artificial datasets designed to help understand the performance, bias and limits of each discretization method.

4.2. The evaluated methods

The discretization methods studied in the comparison are:

- MODL
- MDLPC (Fayyad & Irani, 1992)
- BalancedGain (Kononenko, Bratko, & Roskar, 1984)
- Fusinter (Zighed et al., 1998)
- Khiops (Boullé, 2003, 2004)
- ChiMerge (Kerber, 1991)
- ChiSplit (Bertier & Bouroche, 1981)
- Equal Frequency
- Equal Width

The MDLPC method is a greedy top-down split method, whose evaluation criterion is based on the Minimum Description Length Principle (Rissanen, 1978). At each step of the algorithm, the MDLPC evaluates two hypotheses (to cut or not to cut the interval) and chooses the hypothesis whose total encoding cost (model plus exceptions) is the lowest. The BalancedGain method exploits a criterion similar to the GainRatio criterion (Quinlan, 1993): it divides the entropy-based InformationGain criterion by the log of the arity of the partition in order to penalize excessive multisplits. This method can be embedded into a dynamic-programming based algorithm, as studied in Elomaa and Rousu (1999). The Fusinter method is a greedy bottom-up method that exploits an uncertainty measure sensitive to the sample size. Its criterion employs a quadratic entropy term to evaluate the information in the intervals and is regularized by a second term in inverse proportion of the interval frequencies. The Khiops algorithm uses the chi-square criterion in a global manner to evaluate all the intervals of a discretization, in a greedy bottom-up merge algorithm. Its stopping criterion has been enhanced in Boullé (2003) in order to provide statistical guarantees against overfitting: discretizations of independent attributes consist of a single interval with a user defined probability. The ChiMerge method is a greedy bottom-up merge method that locally exploits the chi-square criterion to decide whether two adjacent intervals are similar enough to be merged. The ChiSplit method, comparable to the ChiMerge method, uses a greedy top-down split algorithm.

The MODL, MDLPC and BalancedGain methods have an automatic stopping rule and do not require any parameter setting. For the Fusinter criterion, we use the regularization parameters recommended in Zighed et al. (1998). The Khiops probability parameter is set to 0.95. For the ChiMerge and ChiSplit methods, the significance level is set to 0.95 for the

chi-square test threshold. The Equal Width and Equal Frequency unsupervised discretization methods use a number of intervals set to 10. Since several instances can share the same descriptive value, the actual number of intervals can be less than expected in the case of the two unsupervised methods.

The MDLPC, ChiMerge and ChiSplit criteria are local to two adjacent intervals: these methods cannot be optimized globally on the whole set of intervals. The Khiops criterion is global, but its stopping criterion is based on the statistic of the variation of the criterion during the merge process used in the algorithm. Thus, the Khiops search algorithm cannot be changed. The BalancedGain and the Fusinter method are the only other compared methods with a global criterion. In order to allow a fair comparison, we use exactly the same search algorithm (greedy bottom-up heuristic followed by the post-optimization) both for the MODL, BalancedGain and Fusinter methods.

4.3. Real data experiments

In this section, we test the evaluated methods on real data. After introducing the datasets and the details of the evaluation protocol, we comment the aggregated results using a multi-criteria analysis. Then, we exhibit the relative differences between the methods on the complete set of experiments. Finally, we illustrate some sample discretizations using histograms.

4.3.1. The datasets

We gathered 15 datasets from U.C. Irvine repository (Blake & Merz, 1998), each dataset has at least one continuous attribute and at least a few tens of instances for each class value in order to perform reliable tenfold cross-validations. Table 3 describes the datasets; the last column corresponds to the relative frequency of the majority class.

The discretizations are performed on the 181 continuous attributes of the datasets, using a ten times stratified tenfold cross-validation. We have re-implemented the alternative discretization methods in order to eliminate any variance resulting from different

Table 3 Datasets

Dataset	Continuous attributes	Nominal attributes	Size	Class values	Majority class
Adult	7	8	48842	2	76.07
Australian	6	8	690	2	55.51
Breast	10	0	699	2	65.52
Crx	6	9	690	2	55.51
German	24	0	1000	2	70.00
Heart	10	3	270	2	55.56
Hepatitis	6	13	155	2	79.35
Hypothyroid	7	18	3163	2	95.23
Ionosphere	34	0	351	2	64.10
Iris	4	0	150	3	33.33
Pima	8	0	768	2	65.10
SickEuthyroid	7	18	3163	2	90.74
Vehicle	18	0	846	4	25.77
Waveform	21	0	5000	3	33.92
Wine	13	0	178	3	39.89

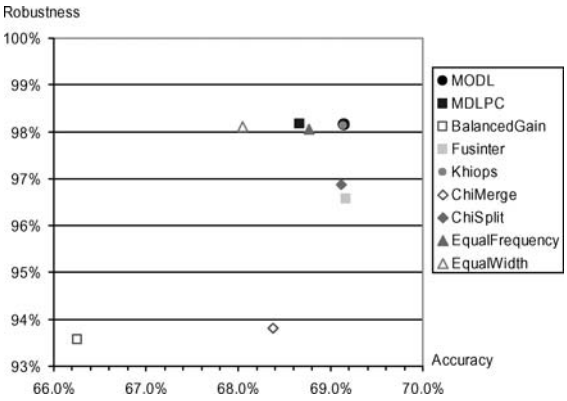


Fig. 2 Bi-criteria evaluation of the discretization methods for the accuracy and the robustness, using datasets geometric means

cross-validation splits. In order to determine whether the performances are significantly different between the MODL method and the alternative methods, the *t*-statistics of the difference of the results is computed. Under the null hypothesis, this value has a Student's distribution with 99 degrees of freedom. The confidence level is set to 1% and a two-tailed test is performed to reject the null hypothesis.

4.3.2. Multi-criteria analysis of the results

The whole result tables related to the 181 attribute discretizations are too large to be printed in this paper. The results are summarized by datasets in the Appendix. Table 10 reports the geometric mean of the accuracy for all the attributes in each dataset, and the global summary by attribute and by dataset. Table 11 details the number of MODL significant wins and losses for each dataset and for all the attributes. Although these three global indicators look consistent, the summary by dataset seems preferable since it gives the same weight to each dataset whatever their number of attributes is. The robustness results are presented in Tables 12 and 13, and the number of intervals results in Tables 14 and 15.

In multi-criteria analysis, a solution *dominates* (or is *non-inferior* to) another one if it is better for all criteria. A solution that cannot be dominated is *Pareto optimal*: any improvement of one of the criteria causes a deterioration on another criterion. The *Pareto surface* is the set of all the Pareto optimal solutions.

In order to analyze both the accuracy and robustness results, we report the dataset geometric means on a two-criteria plan in Fig. 2, with the accuracy on the *x*-coordinate and the robustness on the *y*-coordinate. Similarly, we report the accuracy and the number of intervals in Fig. 3. Each point in these Figures represents the summary of 18,100 experiments. An inspection of the results tables presented in the Appendix shows that a relative difference of about 0.5% between two summary points reflects significant differences between the two corresponding methods. The multi-criteria figures are thus reliable and informative: they allow us to clearly differentiate the behavior of almost all the methods.

The accuracy is the most common evaluated criterion. Looking at this criterion shows that the four methods MODL, Fusinter, Khiops and ChiSplit perform equally well. They are followed by the MDLPC and EqualFrequency methods in a second group. Last come the ChiMerge method, the EqualWidth method and finally the BalancedGain method.

The robustness is an interesting criterion that allows to estimate whether the performance on the train data is a good prediction of the performance on the test data. The higher is the

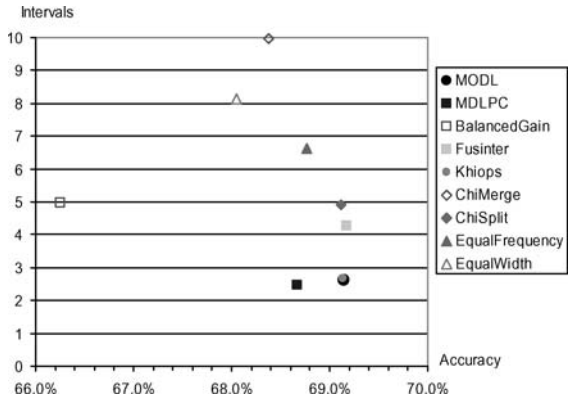


Fig. 3 Bi-criteria evaluation of the discretization methods for the accuracy and the number of intervals, using datasets geometric means

robustness, the most accurate will be a ranking of attributes based on their train accuracy. This can be critical in the case of classifiers such as decision trees that incorporate an attribute selection method to build the next node of the tree. The unsupervised methods exhibit a very good robustness in spite of their large number of intervals. They are not subject to over-fitting since they explore one single hypothesis to discretize the continuous attributes. The MODL, MDLPC and Khiops methods are as robust as the unsupervised methods. They are followed by the Fusinter and ChiSplit methods in a second group. At last, the BalancedGain and ChiMerge methods come with a far weaker robustness.

The number of intervals criterion is potentially related to the simplicity of the discretizations. In the case of decision trees, discretizations with the same accuracy but with fewer intervals are preferable since they cause less fragmentation of the data in the sub-nodes of the tree. The MODL, MDLPC and Khiops methods clearly dominate the other methods on this criterion. Then come the BalancedGain, Fusinter and ChiSplit methods with about twice the number of intervals than that of the leading methods, followed by the unsupervised EqualFrequency and EqualWith methods and finally the ChiMerge method.

If we take into account the three criterion together, the BalancedGain and ChiMerge methods are clearly dominated by all the other methods, followed by the EqualWith method. The EqualFrequency and MDLPC methods obtain similar results on accuracy and robustness, but the MDLPC methods produces far less intervals. Among the four most accurate methods, the Fusinter and ChiSplit methods are clearly dominated both on robustness and number of intervals. The MODL and Khiops methods are Pareto optimal: they dominate all the other methods on the three evaluated criteria. However, they cannot be distinguished from each other.

4.3.3. Detailed differences between the methods

In order to analyze the relative differences of accuracy for the 181 attributes in more details, we collect all the geometric mean ratios per attribute in ascending order. Figure 4 shows the repartition function of the relative differences of accuracy between the MODL method and the other discretization methods. Each point in this repartition function is the summary of 100 discretization experiments performed on the same attribute. The robustness results are presented in Fig. 5 and the number of intervals results in Fig. 6.

Such repartition functions represent a convenient tool for the fine grain analysis of the differences between methods, in complement with the multi-criteria analysis carried out on

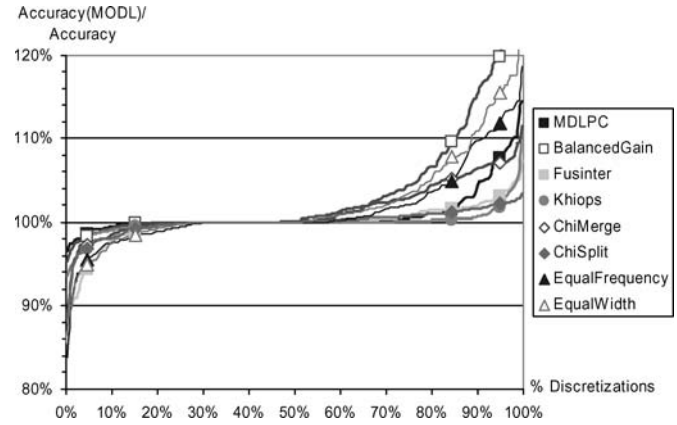


Fig. 4 Repartition function of the relative differences of accuracy between the MODL method and the other discretization methods

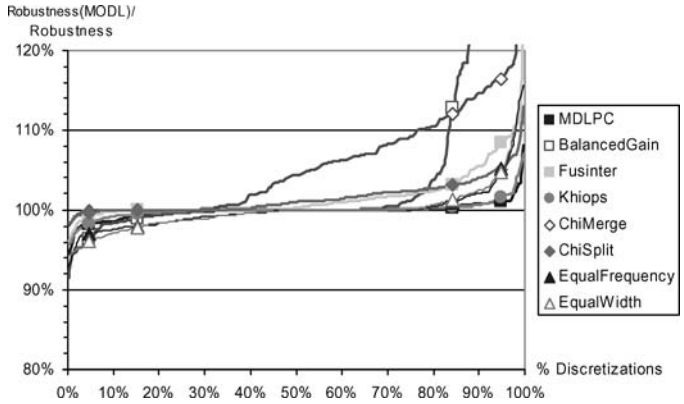


Fig. 5 Repartition function of the relative differences of robustness between the MODL method and the other discretization methods

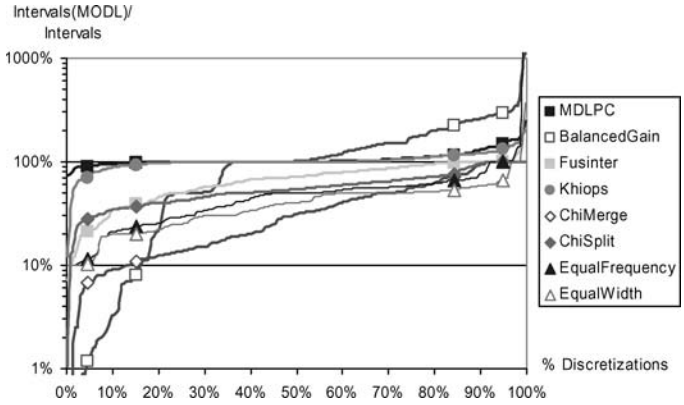


Fig. 6 Repartition function of the relative differences of number of intervals between the MODL method and the other discretization methods

the coarse dataset geometric means. A flat curve reflects two methods that do not differentiate on any of the experiments. A symmetric curve correspond to methods that globally perform equally well, but with differences among the experiences. An unbalanced curve reveals a situation of dominance of one method over the other, with insights on the intensity of the domination and on the size of the region of dominance. On the left of Figs. 4 and 5, the MODL method is dominated by the other methods and, on the right, it outperforms the other algorithms. It is the reverse situation for Fig. 6.

Concerning the accuracy criterion presented in Fig. 4, the Khiops curve is almost flat, with about 80% of the attributes having exactly the same performances than the MODL method. The two other most accurate Fusinter and ChiSplit methods exhibit balanced but slightly dissymmetric curves: they dominate the MODL method in about 10% of the attributes and are dominated—less significantly—in about 20% of the attributes. Compared to the MDLPC method, the MODL method is between 0 and 3% less accurate in about 10% of the attributes, but is between 3 and 10% more accurate in about 10% of the attributes. The average relative difference of 0.7% between the MODL and MDLPC methods is thus significant and reflects potential large differences of accuracy on individual attributes. The other methods are far less accurate than the MODL method on a large proportion of the attributes.

Dealing with the robustness pictured in Fig. 5, the MODL, MDLPC and Khiops methods demonstrate the same level of performance on almost all the attributes. The two unsupervised EqualFrequency and EqualWith methods have the same average level of robustness that the three preceding methods. However, they faintly dominate on 40% of the attributes and are more strongly dominated on 20% of the attributes. The two accurate Fusinter and ChiSplit methods are significantly dominated on a large proportion of the attributes. Finally, the remaining BalancedGain and ChiMerge methods are strongly dominated. The BalancedGain exhibits a very sharp transition for the right-most 20% of the discretizations. In the last 10% (not shown in the figure for scalability reasons), the robustness of the MODL method goes between 25 and 60% higher than that of the BalancedGain method.

Regarding the number of intervals displayed in Fig. 6, the three robust MODL, MDLPC and Khiops methods express similar behaviour on a large majority of the attributes. All the other curves are heavily asymmetrical. The methods produce from 2 to 4 times more intervals on average, and up to one hundred times more intervals in the extreme cases. Once more, the BalancedGain curve distinguishes from the other curves with significantly smaller number of intervals that the MODL method in half of the discretization and the largest numbers of intervals for 15% of the attributes.

The atypical behavior of the BalancedGain method requires some explanation. The BalancedGain criterion is the ratio of the InformationGain by the log of the number of intervals. These two functions are increasing functions with respect to the number of intervals. According to their mutual position, the shape of the ratio function exhibit radically different behaviors. In the case of random or very noisy attributes, the InformationGain is almost null for small numbers of intervals and steadily increases toward its maximum value. The resulting BalancedGain ratio is an increasing function of the number of intervals, with subsequent discretizations containing very large numbers of intervals and having very poor robustness. On the opposite case of very informative attributes, the first split brings a lot of information. This translates the InformationGain first values upward and flatten the increase rate of the function. The ensuing BalancedGain ratio turns into a decreasing function of the number of intervals, with resulting discretizations having only two intervals and a very good robustness. This is equivalent to performing a binary-split using the InformationGain function. Between these two extreme situations, any behavior can theoretically be observed: this happens to be infrequent in the UCI datasets.

Table 4 Mean and standard deviation discretization results for the V1 attribute of the Waveform dataset

	Accuracy		Robustness		Intervals	
Method	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
MODL	33.9	±0.1	100.0	±0.3	1.0	±0.0
MDLPC	33.9	±0.1	100.0	±0.3	1.0	±0.0
BalancedGain	34.6	±2.0	70.1	±4.4	431.0	±6.7
Fusinter	33.8	±1.8	91.9	±5.3	8.3	±1.1
Khiops	33.9	±0.1	100.0	±0.3	1.0	±0.0
ChiMerge	33.7	±2.0	86.0	±5.6	40.2	±5.4
ChiSplit	33.4	±1.0	95.5	±4.1	5.0	±2.8
EqualFrequency	33.1	±1.9	93.0	±5.7	10.0	±0.0
EqualWidth	33.1	±1.9	94.4	±5.8	10.0	±0.0

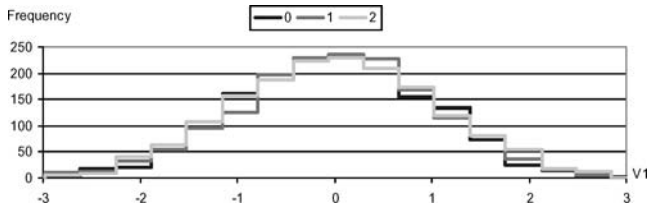


Fig. 7 Class frequency histogram for the V1 attribute of the Waveform dataset

The BalancedGain criterion, very similar to the GainRatio criterion used in C4.5 (Quinlan, 1993), has been experimented in the context of decision trees in Elomaa and Rousu (1999). The reported prediction accuracy results are reasonably good, even though the multi-split discretizations produced by the BalancedGain method are not convincing. Indeed, the selection module of the decision tree always prefers the most informative attributes, which benefit from a very good binary-split (though a weak multi-split) and ignores the noisy attributes which suffer from severe over-splitting. Thus, the complex machinery of a classifier can sometimes hide the effect of poor quality multi-split discretizations.

4.3.4. Some sample discretizations

In this section, we select four attributes among the 181 benchmark attributes, both to illustrate sample discretizations through histograms and to focus on qualitative differences between the evaluated methods.

4.3.4.1. V1 attribute of the waveform dataset. This attribute corresponds to the least predictive attribute among the Waveform attributes. Figure 7 presents the waveform class frequency histogram using 20 bins, computed on the whole dataset.

The three classes look equidistributed, meaning that the V1 attribute holds little conditional information about the class density. The results obtained by the discretization methods during the ten times tenfold cross-validation are reported in Table 4. The three robust MODL, MDLPC and Khiops methods build exactly one interval.

Table 5 Mean and standard deviation discretization results for the V7 attribute of the Waveform dataset

	Accuracy		Robustness		Intervals	
Method	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
MODL	57.5	±1.2	99.9	±2.2	6.0	±0.0
MDLPC	57.4	±1.2	99.9	±2.2	6.0	±0.2
BalancedGain	57.4	±1.2	99.9	±2.2	2.0	±0.0
Fusinter	57.4	±1.2	99.8	±2.3	6.4	±0.6
Khiops	57.5	±1.1	99.7	±2.2	6.5	±0.9
ChiMerge	56.6	±1.4	95.3	±2.7	55.9	±6.4
ChiSplit	57.3	±1.3	99.1	±2.5	14.8	±2.5
EqualFrequency	57.5	±1.1	100.0	±2.2	10.0	±0.0
EqualWidth	57.0	±1.2	100.0	±2.4	10.0	±0.0

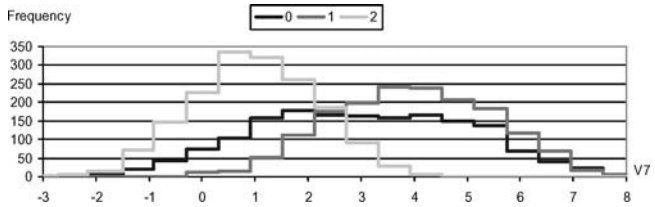


Fig. 8 Class frequency histogram for the V7 attribute of the Waveform dataset

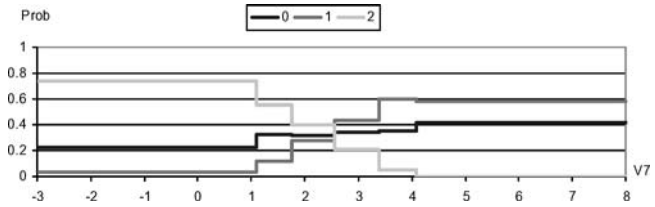


Fig. 9 Class conditional density histogram for the V7 attribute of the Waveform dataset, based on the MODL discretization

4.3.4.2. V7 attribute of the Waveform dataset. This attribute corresponds to the most predictive attribute among the Waveform attributes. Figure 8 presents the class frequency histogram and Table 5 the results of the discretizations.

The MODL methods produces exactly 6 intervals (with a null variance). Figure 9 reports the class conditional density histogram corresponding to the MODL discretization build on the whole Dataset. The discretization behaves like a density estimator, with a focus on the variations of the class densities. The MDLPC, Fusinter and Khiops methods build approximatively the same intervals and obtain the same level of performances on accuracy and robustness. The BalancedGain method produces a binary-split. The other methods builds too many intervals.

4.3.4.3. A30 attribute of the Ionosphere dataset. A close look at the result tables presented in the Appendix indicates a special behavior of the Ionosphere dataset, where the MDLPC method is largely dominated by the other methods. We chose the A30 attribute as one of the

Table 6 Mean and standard deviation discretization results for the A30 attribute of the Ionosphere dataset

	Accuracy		Robustness		Intervals	
Method	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
MODL	80.6	±5.9	97.9	±7.8	5.0	±0.0
MDLPC	73.2	±5.9	99.4	±8.8	3.0	±0.3
BalancedGain	72.4	±6.8	76.0	±7.5	78.3	±20.4
Fusinter	80.5	±5.9	97.8	±7.8	5.0	±0.1
Khiops	80.6	±5.9	97.9	±7.8	5.0	±0.0
ChiMerge	75.5	±7.1	86.3	±8.9	33.2	±5.6
ChiSplit	80.9	±5.6	98.2	±7.5	6.0	±1.0
EqualFrequency	73.3	±7.0	99.8	±10.1	9.0	±0.0
EqualWidth	70.7	±6.5	100.2	±10.2	10.0	±0.0

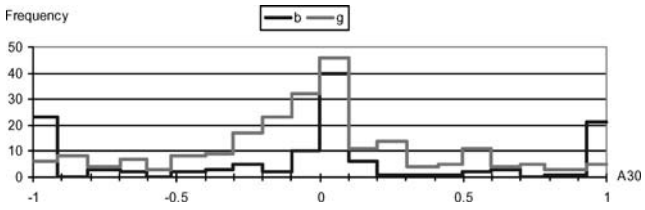


Fig. 10 Class frequency histogram for the A30 attribute of the Ionosphere dataset

most illustrative attributes to explain this behavior. Figure 10 presents the class frequency histogram and Table 6 the results of the discretizations.

The four more accurate methods build between 5 and 6 intervals, whereas the MDLPC methods produces only 3 intervals. Figure 11 shows the class conditional density histogram obtained with the MODL discretization. Two density peaks are identified in the borders of the value domain. These peaks are also identified by the MDLPC method, with exactly the same bounds. The third density peak, in the center, is so thin that it cannot be discovered on the regular class frequency histogram pictured in Fig. 10. Nevertheless, it contains 37 instances and thus corresponds to a reliable class density estimation. This kind of pattern is easily detected by the bottom-up methods, and by the ChiSplit method that tends to produce too many intervals. It is harder to discover for the MDLPC top-down algorithm since it requires two successive splits with the first one not very significant.

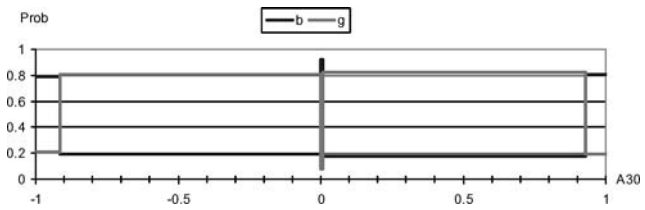


Fig. 11 Class conditional density histogram for the A30 attribute of the Ionosphere dataset, based on the MODL discretization

Table 7 Mean and standard deviation discretization results for the V11 attribute of the Wine dataset

	Accuracy		Robustness		Intervals	
Method	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
MODL	56.8	±7.1	93.4	±14.2	2.9	±0.6
MDLPC	58.8	±8.1	94.9	±14.2	2.8	±0.9
BalancedGain	58.6	±6.7	98.8	±12.1	2.0	±0.0
Fusinter	66.3	±9.9	94.7	±15.7	4.7	±0.7
Khiops	58.7	±6.7	96.2	±13.9	2.3	±0.8
ChiMerge	62.0	±10.4	89.4	±16.7	6.2	±1.1
ChiSplit	62.5	±10.5	91.0	±16.5	6.5	±0.7
EqualFrequency	62.4	±9.8	94.8	±16.1	9.5	±0.5
EqualWidth	63.0	±11.1	97.2	±18.7	9.1	±0.3

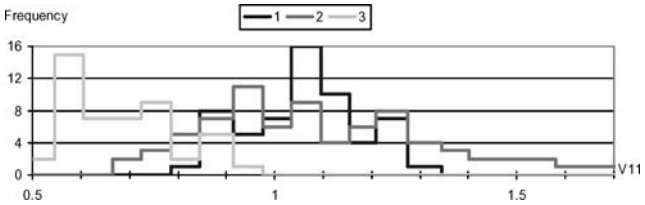


Fig. 12 Class frequency histogram for the V11 attribute of the Wine dataset

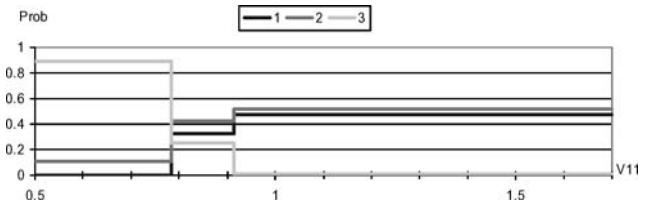


Fig. 13 Class conditional density histogram for the attribute V11 of the Wine dataset, based on the MODL discretization

4.3.4.4. V11 attribute of the wine dataset. This attribute corresponds to one the largest loss in accuracy for the MODL method when compared to the other accurate Fusinter and ChiSplit methods. Figure 12 presents the class frequency histogram and Table 7 the results of the discretizations.

The Wine Dataset contains 178 instances. In a tenfold cross-validation, there are less than 20 instances in the test set: one more correctly classified instance brings a 5% increase in accuracy. This reflects in Table 7 with important differences in accuracy, but large standard deviations. Figure 13 shows that the MODL method builds only three intervals: there are not enough instances to build new reliable intervals.

Figure 14 shows the five intervals of the Fusinter discretization. The last interval is a mixture of classes, since the Fusinter criterion tries to compromise class discrimination and minimum frequency per interval.

Figure 15 shows the seven intervals of the ChiSplit discretization. The class densities are more contrasted than with the Fusinter method, but the methods suffers from over-fitting.

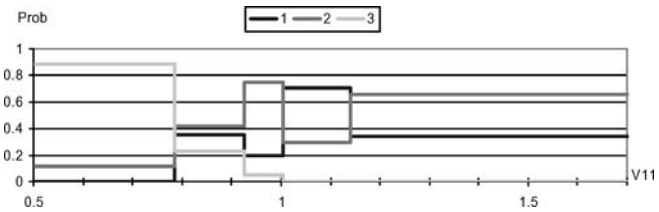


Fig. 14 Class conditional density histogram for the V11 attribute of the Wine dataset, based on the Fusinter discretization

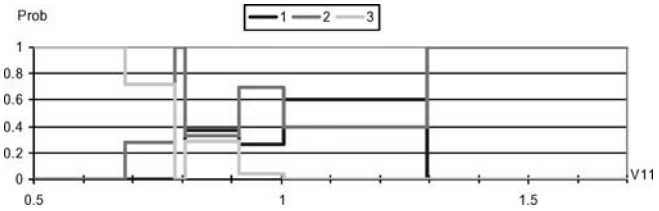


Fig. 15 Class conditional density histogram for the V1 attribute of the Wine dataset, based on the ChiSplit discretization

4.4. Synthetic data experiments

In this section, we focus on the tested supervised discretization methods and try to characterize their bias and performances. We use synthetic datasets, which allow comparing the tested discretization methods with an oracle that knows the true class distribution.

4.4.1. Noise pattern

The purpose of the noise pattern experiment is to evaluate the noise resistance of the discretization methods, under variation of the sample size. In the case of pure noise data, the oracle discretization builds a single interval, meaning that there is no class information in the explanatory attribute. This experiment is strongly biased in favor of the top-down discretization heuristics, since they just have to reject one single split to be as good as the oracle. On the contrary, the bottom-up methods have to accept many merge decisions before they can produce a single interval discretization.

The *noise pattern* dataset consists of an explanatory continuous attribute independent from the class attribute. The explanatory attribute is uniformly distributed on the [0, 1] numerical domain and the class attribute consists of two equidistributed class values. The evaluated criterion is the number of unnecessary intervals, since the test accuracy is always 0.5 whatever the number of intervals is in the discretizations. The experiment is done on a large number of sample sizes ranging from 100 instances to 100,000 instances. In order to obtain reliable results, it is performed 10,000 times. Figure 16 presents the average unnecessary interval number obtained by the tested discretization methods, for different sample sizes.

The BalancedGain, ChiMerge and ChiSplit methods always produce more than one interval, and the number of intervals increases with the sample size. The Fusinter method builds between 5 and 10 intervals with a maximum when the sample size is about 5,000. The Khiops method is designed to produce one single interval with probability greater than 0.95 when the explanatory attribute is independent from the class attribute. This is confirmed

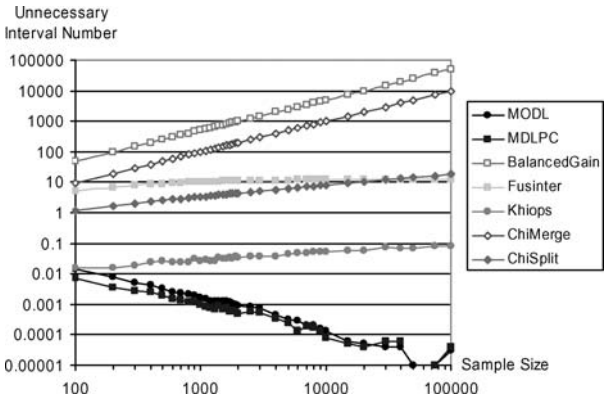


Fig. 16 Mean of the unnecessary interval number of the discretizations of an explanatory attribute independent from the class attribute

by the experiment with an average unnecessary interval number ranging from 0.02 to 0.08 (even a 0.1 unnecessary interval number corresponds to an average 5% of multi-interval discretizations, since most of these discretizations consist of 3 intervals). The MODL and MDLPC method are very resistant to noise and almost always produce a single interval. The experiments have been performed 100,000 times for these two methods in order to better approximate the result curves in Fig. 16. The percentage of multi-interval discretization is about 1% of the cases for sample size 100; it decreases with the sample size with an approximate rate of $1/n$. This behavior seems consistent since the probability of finding an information pattern in a randomly generated attribute decreases with the sample size. In the few cases of multi-interval discretization, the MDLPC always constructs two intervals, since it is strongly biased by its top-down algorithm in favor of information patterns in the borders of the numerical domain. On the opposite, the MODL method builds discretization containing either two or three intervals, since its bottom-up heuristic allows to identify information patterns surrounded by two noise patterns. This behavior explains why the average unnecessary interval number is slightly larger for the MODL method than for the MDLPC method.

4.4.2. Crenel pattern

The crenel pattern experiment is based on a more complex dataset family, built with a "crenel pattern". The explanatory attribute is uniformly distributed on the [0, 1] numerical domain and the class attribute consists of two equidistributed class values '+' and '−' which alternate on the explanatory value x according to the function $Class = Sign(Sinus(100\pi x))$. In a noise free context, the optimal discretization consists of 100 intervals whose bounds are equidistant on the [0, 1] numerical domain. For small sample sizes, where the number of instances is comparable to the optimal interval number, the distribution of the class values on the numerical domain is similar to noise. In this case, the discretization methods should produce a single interval, related to a test error of 0.5. When the sample size increases, each optimal interval contains more and more instances and can correctly be identified by the discretization methods. Thus, the test error asymptotically decreases towards 0 when the sample size increases. The experiment points out the threshold of the sample size above which all the optimal intervals are correctly detected.

The experiment is performed for a large number of sample sizes ranging from 100 instances to 100,000 instances. The evaluated criterion is the mean of the number of intervals on 1,000 randomly generated samples, for each sample size. We can notice that the 1R discretization

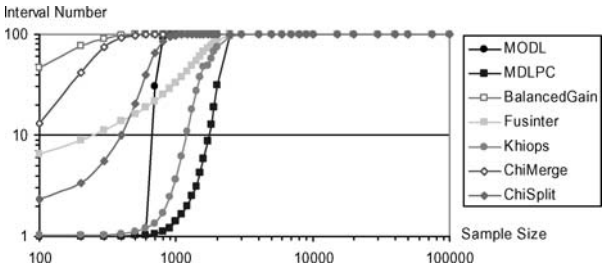


Fig. 17 Mean interval number for the discretization of the "crenel pattern" $Class = Sign(Sinus(100\pi x))$ on the numerical domain [0, 1], in a noise free context

method (Holte, 1993) which produces one interval for each sequence of instances belonging to the same class is optimal for this noise-free problem. The results for the tested discretization methods are displayed in Fig. 17.

The BalancedGain and ChiMerge methods which produce many intervals (cf. UCI experiments) are favored in this noise free context: they are the first methods that correctly identifies the 100 intervals, with about 500 instances. The Fusinter, ChiMerge and ChiSplit methods overfit the data with too many intervals for very small sample sizes. On the opposite, the MODL, Khiops and MDLPC methods produce one single interval for small sample sizes. The transition between 1 interval and 100 intervals happens very sharply at sample size 700 for the MODL method. The MDLPC, Fusinter and Khiops methods need between 2,000 and 3,000 instances to identify all the intervals.

The same experiment is performed in a noisy context, with 10% of randomly misclassified instances. The results are displayed in Fig. 18. Once again, the BalancedGain, Fusinter, ChiMerge and ChiSplit methods overfit the data with too many intervals, but the Fusinter method does not asymptotically build more than the necessary number of intervals. The BalancedGain method, whose criterion compromises the InformationGain and the number of intervals owing to a ratio, displays an abrupt change in its behavior beyond 30000 instances. The ChiSplit method produces several times the number of necessary intervals when the sample size is large. However, the difference in test accuracy (shown in Fig. 19) between the ChiSplit method and the robust methods is asymptotically small. The robust methods MODL, MDLPC and Khiops produce one single interval for small sample sizes and identify the correct intervals for large sample sizes. The transition is still abrupt for the MODL method, at sample size 1,000, whereas it is smoother for the other methods. The correct discretization is found at sample size 3,000 for the Khiops method, 4000 for the Fusinter method and 7,000 for the MDLPC method. When the sample size increases, the MDLPC method slightly overfits the data, with discretizations containing about 103 intervals for sample size 100,000.

To summarize, the MODL method is both robust and sensitive. It produces as few intervals as the most robust methods when the sample size is small, and as many interval as necessary once there is enough instances in the sample.

4.4.3. Peak pattern

The peak pattern experiment focuses on the detection of a pure interval (with instances belonging to one single class) hidden in the center of a noise attribute. In order to prevent superfluous over-splitting, the explanatory attribute consists of only three values: one for the first noisy interval, one for the center pure interval and one for the last noisy interval. The class attribute is composed of two classes. The pure interval contains only instances of the first class and the other intervals share the remaining instances with the same class proportion. The experiment points out the threshold of the peak interval size above which the pattern

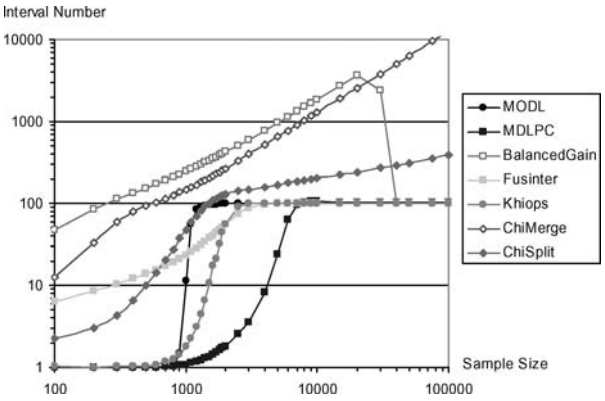


Fig. 18 Mean interval number for the discretization of the "crenel pattern" $Class = Sign\,(Sinus\,(100\pi x))$, with 10% misclassified instances

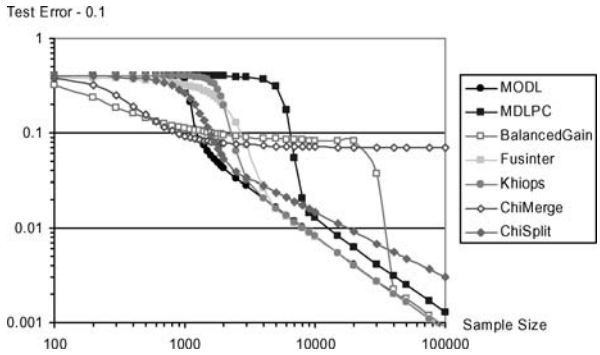


Fig. 19 Mean distance to the true test error for the discretization of the "crenel pattern" $Class = Sign\,(Sinus\,(100\pi x))$, with 10% misclassified instances

is correctly identified. Three intervals are necessary when the peak is in the center, and two intervals when the peak is in the head.

In order to get an idea of the frequency of such patterns in randomly distributed attributes, we propose below an approximation of such a threshold. Let p_1 and p_2 be the two class probabilities, n the number of instances in the sample and k the size of the peak interval. The probability that a sequence of k instances contains only instances of the first class is p_1^k. If $k \ll n$, there are about n such sequences in the sample. Although they intersect on a small scale, we assume that they are independent for approximation reasons. The probability of observing no pure sequence of k instances in the sample is:

$$\left(1 - p_1^k\right)^n .$$

Thus, observing at least one pure sequence of length k in a random sample of size n becomes probable beyond $k \sim -\log(n)/\log(p_1)$.

Figure 20 displays the threshold of the pure interval size for the evaluated discretizations under variation of the sample size, when the two classes are equidistributed. The UnbalancedGain method always produces multi-split discretizations, even when there is one single instance in the peak interval. The ChiMerge and ChiSplit methods are identical when the peak is in the head: they overfit the data. When the peak is in the center, the ChiSplit method suffers from its top-down algorithm and requires an increasing number of instances with the sample size: it needs about 10 times the theoretical peak size approximation for sample size 10000. The Fusinter method exploits a regularization technique that penalizes intervals with

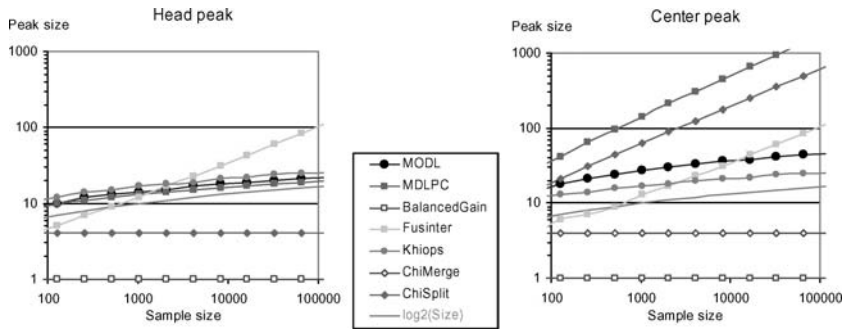


Fig. 20 Threshold for the detection of a pure peak interval hidden into a noise attribute, under variation of the sample size

small frequency. The penalty is heuristically adjusted, resulting in an overfitting behavior for small sample sizes and an underfitting behavior for larger sample sizes, wherever the peak is located. The MODL, MDLPC and Khiops methods exhibit a very similar behavior in the case of a head peak, quite close from the theoretical threshold approximation. When the peak is in the center, the MDLPC methods has the same drawback that the ChiSplit method, due to its top-down approach. The Khiops methods has exactly the same behavior for head and center peaks, whereas the MODL methods requires slightly larger peaks in the center, when more complex discretizations (with three intervals instead of two) are necessary.

We exploit the same experiment to study the asymptotic behavior of the methods, when the class distribution becomes more and more unbalanced. The Khiops, ChiMerge and ChiSplit methods are based on chi-square statistics. The chi-square value is not reliable to test the hypothesis of independence if the expected frequency in any cell of the contingency table is less than some minimum value. This is equivalent to a minimum frequency constraint for each row of the contingency table. The tuning of this constraint is a compromise between reliability and fine pattern detection: no optimal trade-off can be found.

Figure 21 displays the threshold of the pure interval size for the evaluated discretizations under variation of the probability of first class, for a sample size fixed to 10000 instances. In addition to the theoretical threshold approximation, we report the chi-square theoretical minimum interval size that corresponds to an expected frequency of at least 1 instance in each cell of the contingency table. The most notable result is a quasi-identical behavior of the MODL and MDLPC methods in the case of head peaks, very close from the theoretical threshold approximation. The chi-square based methods ChiMerge and ChiSplit do no exploit a minimum frequency constraint: their use of the chi-square statistics is not reliable as soon as the first class probability is lower than 0.1. The Khiops method heuristically incorporates the minimum frequency constraint: it is still both too aggressive to get a reliable use of the chi-square statistics and to conservative to allow fine grain detection when the class distribution is unbalanced. The top-down based algorithms MDLPC and ChiSplit fail to correctly identify pure peaks when they are located in the center. The MODL method obtains the finest results for peak detection wherever they are located. In extremely unbalanced class distributions, the MODL method requires only two instances in head pure peak intervals and fours instances in center pure peak intervals.

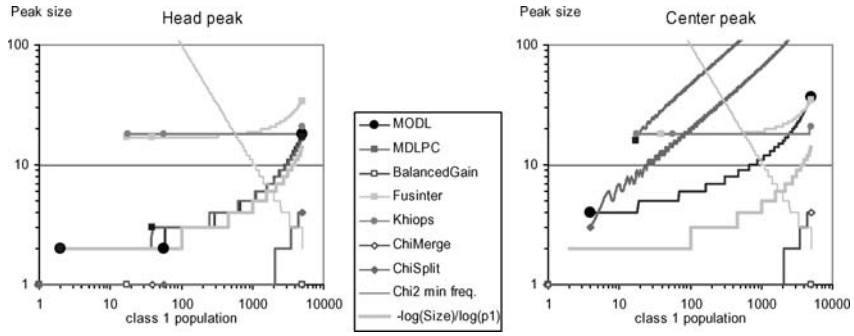


Fig. 21 Threshold for the detection of a pure peak interval hidden into a noise attribute, under variation of the first class population, for sample size 10000

4.4.4. Discussion

The experiments on synthetic data are a way of characterizing each discretization methods and its bias. Some of these methods are very robust to noise, some of them are very accurate in noise free context. It is interesting to see that the widely used accuracy criterion is not suitable to precisely evaluate and compare different inductive methods. It does not penalize the methods which build patterns from noise. Furthermore, the methods which overfit the data have a minor penalty when the detected pattern is close to noise and a large reward when it is a true pattern. Finally, the accuracy criterion does not clearly favor the methods which are more and more accurate as the sample size increases. The difference in accuracy is often small while the difference in correctness of the model might be important. These synthetic experiments are a first step towards a better evaluation of the robustness of the discretization methods and their ability to find complex patterns even in a noisy context. There is still a need for a more complete set of synthetic benchmarks and perhaps for the design of new criteria in order to consistently evaluate the inductive methods on real datasets.

Overall, the synthetic experiments enlighten a large diversity of discriminating behaviors among the evaluated methods. The two MDLPC and Khiops methods are the closest from the MODL method. Compared with the MDLPC method, the MODL method extends the recursive binary-split approach towards a true multi-split discretization schema and largely augments the capacity of fine grain pattern detection. Compared with the Khiops method, the MODL method does not require any parameter for the level of resistance to noise and removes the asymptotical limitations caused by the use of chi-square statistics.

5. Optimization criterion versus search strategy

In this section, we study the relative contribution of the optimization criterion versus the search strategy to the quality of the discretizations. We first focus on computation time and its relation to the numerical efficiency of the optimization. We then examine the connection between the criterion optimality and the discretizations quality.

5.1. Computation times of the evaluated algorithm

The evaluated discretization methods have the same computational complexity of $O(n \log(n))$. They first sort the attribute values and identify boundary instances in a preprocessing step.

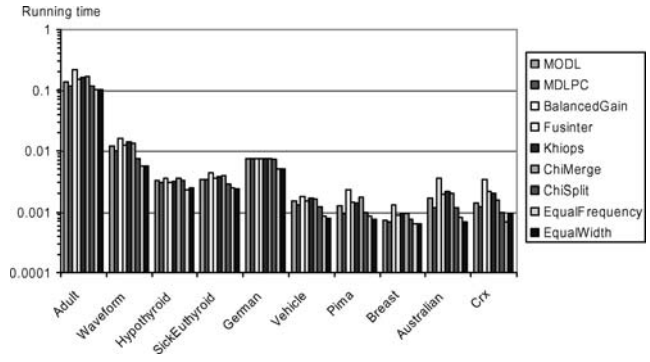


Fig. 22 Computation time of the discretization methods on ten UCI datasets

The time efficiency of the methods mainly relies on the search direction (top-down or bottom-up), the simplicity of the mathematical criterion and the use of a post-optimization algorithm. The actual running time on real datasets depends on the final number of intervals found in the attribute discretizations.

The UCI dataset experiments are conducted on a PC with a Pentium IV 2.5 Ghz, using the ten times tenfold cross-validation protocol. Figure 22 reports the average discretization running time per attribute (in seconds) for the ten larger datasets, ranging from 50000 instances on the left to 700 instances on the right. All the supervised discretization methods obtain comparable running times. They are on average between 50 and 150% longer than the unsupervised discretization methods.

5.2. Computation time versus criterion optimality

The MODL, BalancedGain and Fusinter methods are the only evaluated methods whose criterion can be globally optimized. The other methods MDLPC, ChiMerge and ChiSplit use a recursively applied binary-split criterion. The Khiops method exploits a regularization technique that is intrinsically coupled with its search strategy.

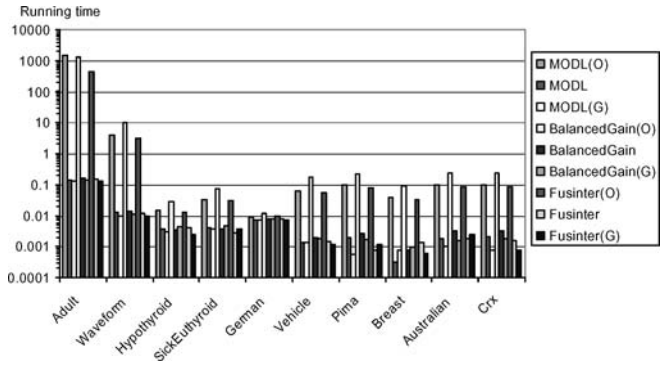


Fig. 23 Computation time of the MODL, BalancedGain and Fusinter discretization methods on ten UCI datasets, using the optimal algorithm (O), the new evaluated algorithm and the greedy algorithm (G)

The UCI experiments are completed with the three MODL, BalancedGain and Fusinter methods, using three search strategies: the greedy bottom-up optimization algorithm, the new evaluated search algorithm that incorporates an additional post-optimization step and the optimal dynamic-programming based algorithm. In order to obtain all the results in less than one week of CPU time, the maximum number of intervals is set to 100 for all the methods and the tenfold cross-validation is performed just one single time. Figure 23 presents the average discretization running times per attribute. The results show a consistent behavior of the algorithms for the three criteria. The post-optimization algorithm presents a 20% running time overhead compared to the greedy algorithm. As expected, the optimal algorithm, whose computational complexity is $O(n^3)$, requires much more running time than the two search heuristics.

We report in Table 8 the efficiency of the three search strategies, in terms of their ability to reach the optimal solution or to improve the relative distance to the optimal solution. The "Time" column in Table 8 stands for the geometric mean (by dataset) of the computation time normalized by the Equal Frequency computation time. For example, the MODL method with the post-optimization algorithm takes on average 1.7 more computation time than the Equal Frequency algorithm. The "Diff" column represents the mean (by attribute) of the relative distance of the method criterion to the optimal solution $(criterion_{\text{algorithm}} - criterion_{\text{optimal}})\ /criterion_{\text{optimal}}$. The "% opt" column presents the percentage of optimal solution found by the algorithm.

The results are still consistent for the three evaluated criteria. The post-optimization algorithm reaches the optimal solution about twice as many times than the greedy algorithm and the relative distance to the optimal solution is improved by a factor 100. From a strict optimization point of view, the post-optimization algorithm radically improves the results of the greedy algorithm with a small computational overhead.

Table 8 Dataset geometric mean of the normalized computation time (Time), mean of the relative distance to the optimal solution (Diff) and percentage of optimal solution (% opt) for the MODL, BalancedGain and Fusinter discretization methods, using three search strategies

	MODL			BalancedGain			Fusinter		
Algorithm	Time	Diff	% opt	Time	Diff	% opt	Time	Diff	% opt
Optimal	82.3	0.0	100%	161.7	0.0	100%	64.0	0.0	100%
Post-opt	1.7	8.6E − 5	95%	2.2	2.4E − 3	90%	1.7	8.3E − 5	91%
Greedy	1.3	3.8E − 3	55%	1.9	1.3E − 1	41%	1.4	4.0E − 3	40%

Table 9 Dataset geometric means of the accuracy, robustness and number of intervals for the MODL, BalancedGain and Fusinter discretization methods, using three search strategies

	MODL			BalancedGain			Fusinter		
Algorithm	Accur.	Robus.	Int.	Accur.	Robus.	Int.	Accur.	Robus.	Int.
Optimal	69.14	98.17	2.62	66.42	94.83	4.12	69.11	96.43	4.37
Post-opt	69.18	98.23	2.60	66.48	94.84	4.19	69.16	96.57	4.27
Greedy	69.06	98.09	2.80	66.01	90.18	11.04	68.99	96.52	4.29

5.3. Impact on the quality of the discretizations

Reaching the optimal value of a mathematical criterion does not mean finding the best discretization. The notion of best discretization is hard to define: we restrict this to the evaluation of the accuracy, robustness and number of intervals of the discretizations, as reported in Table 9. The optimal and post-optimized search strategies obtain statistically the same results: there are only 5 significant differences (not reported here) between the two search strategies among 1629 individual comparison experiments (181 attributes, 3 evaluations, 3 criteria). Compared to the greedy search strategy, they both bring a slight enhancement in accuracy and robustness, and a more significant reduction in the number of intervals. The Fusinter criterion is the less sensitive one to the quality of the optimization. The BalancedGain criterion highly benefits from the optimization by escaping from local optima with numerous intervals in the case of informative attributes.

Overall, the evaluated quality of the discretizations is improved by better search strategies for all the criteria. However, the criterion is far more important than the search strategy to obtain high quality discretizations.

5.4. Impact on the quality of the explanation

In the case of the MODL criterion, we can propose an interpretation of the improvements in the optimized criterion owing to Theorem 5. The absolute difference between the evaluation of a discretization and the optimal evaluation is directly connected to the probability that the discretization explains the data.

Theorem 5. *Let M be a MODL discretization of a sample data D, Cost(M) its evaluation with the MODL criterion. Then optimal discretization* M_{opt} *is a more probable MODL explanation of D than M the with the following factor:*

$$\exp(Cost\,(M) - Cost(M_{\text{opt}})).$$

Proof: The probability that M explains D is $p(M/D)$. According to the proof of Theorem 1, this probability is related to the MODL evaluation cost by:

$$Cost(M) = -\log(p(M/D)p(D))$$

Thus,

$$\log\left(\frac{p(M_{\text{opt}}/D)}{p(M/D)}\right) = Cost(M) - Cost(M_{\text{opt}}).$$

The claim follows. □

In Fig. 24, we report the distance to the optimal discretization for the 1810 discretization performed on the UCI datasets. For example, with the greedy heuristic, 20% of the discretizations are at least 10 times less probable than the optimal discretization to explain the data. The quality of the discretization-based explanation of the data is thus significantly improved by the post-optimization algorithm.

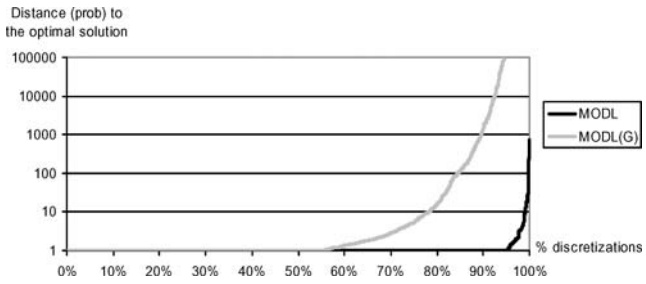


Fig. 24 Repartition function of the distance to the optimal discretization evaluated on 1810 discretizations, for the new evaluated algorithm and the greedy heuristic (G)

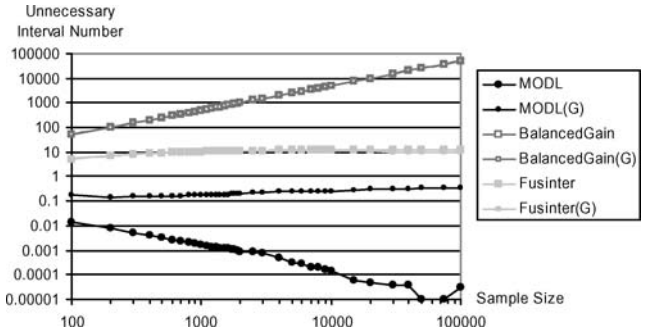


Fig. 25 Mean of the unnecessary interval number of the discretizations of an explanatory attribute independent from the class attribute, for the post-optimized search strategy and the greedy algorithm (G)

We finally compare the two search heuristics in the case of the noise pattern experiment. The optimal algorithm is not applicable due to the size of the datasets and to the number of repetitions of the experiments. Figure 25 shows no differences between the two search strategies for the BalancedGain and Fusinter criteria. On the opposite, the change of behavior is very important for the MODL criterion. The greedy algorithm produces on average 0.2 unnecessary intervals instead of almost exactly the expected number of intervals for the post-optimized algorithm.

This last experiments enlightens the main conclusions of this section. The criterion clearly comes first, before the search strategy. The impact of better optimized discretization is small on common evaluations such as accuracy, robustness or number of intervals. It is still important when the quality of the explanation of the data is considered.

6. Conclusion

The MODL discretization method takes advantage of the precise definition of a family of discretization models with a general prior. This provides a new evaluation criterion which is minimal for the Bayes optimal discretization, i.e. the most probable discretization given the data sample. A new optimization heuristic is proposed in this paper to optimize the discretization with super-linear time complexity. This algorithm allows to find the optimal discretization in most cases.

Extensive evaluations both on real and synthetic data enlighten the key features of the MODL method. It is time efficient and does not require any parameter setting. It builds discretizations that are both robust and accurate, resistant to noise and sensitive to fine grain patterns. It correctly identifies complex patterns provided that there is enough data, needing less data than the alternative robust methods. It produces fewer intervals than most alternative accurate methods and has no asymptotic limitation.

The most valuable characteristic of the MODL method potentially resides in the robust explanation of the data it provides. Even though this is restricted to choice of its model space and limited by the bias of its model prior, the MODL method builds the most probable discretization-based explanation of the data.

In future work, we plan to extend this approach to the problem of grouping the values of categorical attributes.

Appendix

Table 10 Geometric mean of the accuracy per attribute discretization

Dataset	MODL	MDLPC	BGain	Fusin	Khiops	ChiM	ChiS	EqFr	EqWi
Adult	77.34	77.33	74.07	77.27	77.30	75.63	77.32	76.62	76.76
Australian	64.31	64.39	62.40	63.46	64.42	63.86	64.31	65.14	60.72
Breast	85.35	85.55	85.08	85.40	85.35	85.12	85.56	85.24	85.63
Crx	64.41	64.46	62.39	63.41	64.55	63.45	64.49	64.99	60.62
German	70.05	70.00	69.98	70.03	70.04	69.99	70.02	69.90	70.07
Heart	63.57	63.19	63.26	62.68	63.62	62.77	63.00	64.12	63.09
Hepatitis	79.13	79.15	75.98	79.42	79.52	77.92	78.85	80.15	79.99
Hypothyroid	96.00	96.03	95.20	96.04	96.04	96.02	96.02	95.21	95.40
Ionosphere	79.64	77.37	76.21	79.19	79.40	75.73	79.23	73.93	73.39
Iris	76.86	72.91	60.92	77.68	75.98	75.49	76.54	73.42	74.69
Pima	66.10	65.98	64.94	66.01	66.12	65.97	66.08	66.50	66.31
SickEuthyroid	91.29	91.29	91.24	91.28	91.30	91.28	91.31	91.18	90.72
Vehicle	40.21	39.58	35.72	40.53	40.56	40.52	40.82	39.75	39.95
Waveform	48.71	48.77	48.64	48.57	48.75	47.92	48.51	48.79	48.52
Wine	58.97	58.74	55.44	61.21	58.66	58.26	59.15	60.23	59.48
All attributes	66.24	65.67	63.86	66.28	66.25	65.26	66.22	65.35	64.88
All datasets	69.14	68.67	66.26	69.17	69.14	68.38	69.12	68.77	68.04

Table 11 Number of MODL significant wins and losses for the accuracy criterion

Dataset	MDLPC	BGain	Fusin	Khiops	ChiM	ChiS	EqFr	EqWi
Adult	2/0	5/0	2/0	4/0	3/2	2/2	2/0	3/0
Australian	1/2	2/1	3/1	1/1	2/0	3/2	0/4	5/1
Breast	0/1	2/1	2/2	1/1	2/2	1/2	2/2	1/2
Crx	1/2	2/1	4/1	1/1	4/0	2/2	0/4	5/1
German	1/0	1/0	1/1	0/0	1/0	1/1	2/0	1/2
Heart	1/0	1/0	1/0	0/0	3/0	2/0	2/3	2/1
Hepatitis	0/0	3/1	3/2	0/1	3/3	2/2	0/2	3/3

(*Continued on next page*)

Table 11 (*Continued*)

Dataset	MDLPC	BGain	Fusin	Khiops	ChiM	ChiS	EqFr	EqWi
Hypothyroid	1/2	4/0	1/1	1/2	3/2	2/2	3/0	3/0
Ionosphere	17/2	23/1	14/2	5/5	29/0	16/2	29/0	26/0
Iris	2/0	4/0	0/1	1/1	1/1	1/0	2/0	2/1
Pima	1/1	3/1	2/2	1/1	2/2	1/1	1/2	3/2
SickEuthyroid	0/0	2/0	1/0	0/0	3/2	2/1	1/0	2/0
Vehicle	8/1	14/0	4/5	3/6	4/5	3/7	7/5	6/6
Waveform	2/4	4/9	6/2	1/2	14/0	11/1	3/7	9/5
Wine	4/3	6/2	0/7	5/3	3/1	4/2	1/3	3/5
Total	41/18	76/17	44/27	24/24	77/20	53/27	55/32	74/29

Table 12 Geometric mean of the robustness per attribute discretization

Dataset	MODL	MDLPC	BGain	Fusin	Khiops	ChiM	ChiS	EqFr	EqWi
Adult	99.99	99.99	90.63	100.0	99.96	94.98	99.91	100.0	100.0
Australian	97.12	97.55	84.95	94.18	97.40	92.34	96.10	98.88	99.00
Breast	99.74	99.80	95.20	98.67	99.46	96.91	99.18	99.67	99.69
Crx	97.35	97.68	84.94	94.20	97.55	91.66	96.39	98.65	98.81
German	99.92	99.89	99.66	99.81	99.91	99.71	99.79	99.84	99.86
Heart	98.78	98.43	92.69	94.22	98.85	93.45	96.22	97.50	97.00
Hepatitis	98.40	99.25	90.26	96.71	98.42	92.70	97.11	99.39	98.23
Hypothyroid	99.91	99.91	99.94	99.89	99.86	99.79	99.84	100.0	99.99
Ionosphere	97.57	97.77	90.52	96.66	97.63	87.36	96.06	98.65	98.87
Iris	96.94	94.55	97.14	97.49	96.76	95.54	96.49	94.92	96.27
Pima	99.14	98.92	92.57	97.23	99.04	95.24	97.55	98.92	98.10
SickEuthyroid	99.96	99.96	99.94	99.95	99.94	99.78	99.93	100.0	99.99
Vehicle	94.87	95.16	96.10	92.40	95.40	90.98	92.64	93.82	94.47
Waveform	98.68	98.76	94.30	96.65	98.68	91.99	96.76	98.12	98.57
Wine	94.29	95.30	96.49	91.17	93.31	86.18	89.65	92.84	93.13
All attributes	98.01	98.14	94.03	96.46	98.00	92.83	96.54	97.93	98.06
All datasets	98.16	98.18	93.57	96.58	98.13	93.82	96.87	98.05	98.11

Table 13 Number of MODL significant wins and losses for the robustness criterion

Dataset	MDLPC	BGain	Fusin	Khiops	ChiM	ChiS	EqFr	EqWi
Adult	0/0	2/0	0/0	2/0	3/0	2/0	0/0	0/0
Australian	1/2	3/1	5/1	1/1	5/0	4/1	0/4	0/4
Breast	0/0	2/0	2/0	2/0	2/0	2/0	1/0	1/0
Crx	1/2	3/1	5/1	1/1	5/0	4/1	0/4	1/2
German	1/0	3/0	1/0	0/0	3/0	1/0	1/0	1/1
Heart	1/0	2/0	5/0	0/0	4/0	5/0	2/1	3/0
Hepatitis	0/1	3/0	3/1	1/1	4/0	2/0	0/1	3/1
Hypothyroid	1/0	1/0	1/0	1/0	4/0	2/0	0/1	0/1
Ionosphere	1/5	17/3	17/1	2/4	32/0	25/0	0/12	2/14
Iris	2/0	1/2	0/1	1/1	1/2	1/0	1/2	1/2
Pima	1/1	4/1	4/0	1/0	7/0	5/0	1/0	4/0
SickEuthyroid	0/0	1/0	1/0	0/0	4/0	4/0	0/0	1/0

(*Continued on next page*)

Table 13 (*Continued*)

Dataset	MDLPC	BGain	Fusin	Khiops	ChiM	ChiS	EqFr	EqWi
Vehicle	2/4	2/7	7/1	2/5	12/0	10/0	5/3	3/4
Waveform	2/3	4/9	17/0	2/2	21/0	20/0	5/6	2/8
Wine	1/4	1/5	5/0	5/1	11/0	10/0	2/1	4/5
Total	14/22	49/29	73/6	21/16	118/2	97/2	18/35	26/42

Table 14 Geometric mean of the number of intervals per attribute discretization

Dataset	MODL	MDLPC	BGain	Fusin	Khiops	ChiM	ChiS	EqFr	EqWi
Adult	5.56	5.39	25.05	8.42	6.84	93.47	17.15	4.62	9.29
Australian	2.13	1.95	15.82	6.28	2.12	13.49	4.86	7.74	7.80
Breast	2.58	2.73	3.24	3.80	2.50	5.83	4.51	4.79	8.74
Crx	2.17	1.97	15.99	6.23	2.14	13.39	4.83	7.75	7.80
German	1.18	1.16	2.35	1.94	1.21	1.92	1.80	2.45	3.22
Heart	1.62	1.60	3.75	3.06	1.62	3.38	2.32	4.54	4.79
Hepatitis	1.47	1.35	6.47	3.21	1.51	6.02	2.55	8.08	8.70
Hypothyroid	2.31	2.53	3.77	2.19	3.82	8.52	4.36	6.94	9.60
Ionosphere	4.49	3.65	8.05	5.80	4.13	25.76	7.31	7.73	8.84
Iris	2.99	2.75	2.00	3.10	2.82	3.68	3.60	7.43	9.70
Pima	2.13	1.91	6.67	5.01	2.19	9.50	4.89	8.11	9.43
SickEuthyroid	2.84	2.83	3.06	3.41	3.11	12.79	5.26	6.94	9.59
Vehicle	4.12	3.69	2.20	5.75	3.70	8.48	7.44	8.29	9.57
Waveform	4.37	4.46	4.27	7.83	4.07	48.14	12.20	10.00	10.00
Wine	2.72	2.62	2.01	4.00	2.53	6.15	4.56	9.40	9.83
All attributes	2.79	2.62	4.50	4.40	2.77	10.71	5.18	6.44	7.73
All datasets	2.60	2.48	4.96	4.27	2.68	9.96	4.91	6.63	8.15

Table 15 Number of MODL significant wins and losses for the number of intervals

Dataset	MDLPC	BGain	Fusin	Khiops	ChiM	ChiS	EqFr	EqWi
Adult	1/1	2/5	4/2	3/3	7/0	7/0	2/5	5/2
Australian	0/3	3/2	6/0	2/1	6/0	6/0	6/0	6/0
Breast	4/0	1/8	7/0	1/5	10/0	10/0	10/0	10/0
Crx	0/4	3/2	6/0	1/2	6/0	6/0	6/0	6/0
German	0/4	20/1	17/0	3/0	15/0	15/0	19/1	24/0
Heart	0/1	3/0	6/0	0/0	6/0	6/0	8/0	8/0
Hepatitis	0/1	4/0	6/0	2/0	6/0	6/0	6/0	6/0
Hypothyroid	4/0	2/3	2/3	6/0	7/0	7/0	7/0	7/0
Ionosphere	1/21	14/11	29/0	0/17	32/0	31/0	32/0	32/0
Iris	1/2	0/4	2/0	0/1	4/0	4/0	4/0	4/0
Pima	2/3	4/2	8/0	3/1	8/0	8/0	8/0	8/0
SickEuthyroid	0/1	1/4	5/0	4/0	6/0	6/0	6/0	7/0
Vehicle	0/12	0/16	17/1	4/9	18/0	18/0	18/0	18/0
Waveform	9/2	4/17	21/0	5/13	21/0	21/0	21/0	21/0
Wine	1/5	0/10	12/0	1/7	13/0	13/0	13/0	13/0
Total	23/60	61/85	148/6	35/59	165/0	164/0	166/6	175/2

Acknowledgments I am grateful to the editor Prof. Tom Fawcett and the three anonymous reviewers for their numerous beneficial comments, especially concerning the experimental study and the comparative evaluation of the contribution of the criterion and the search strategy.

References

Bay, S. (2001). Multivariate discretization for set mining. *Knowledge and Information Systems*, *3*(4), 491–512.

Bertier, P., & Bouroche, J. M. (1981). *Analyse des données multidimensionnelles*. Presses Universitaires de France.

Blake, C. L., & Merz, C. J. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

Boullé, M. (2003). Khiops: A discretization method of continuous attributes with guaranteed resistance to noise. *Proceeding of the Third International Conference on Machine Learning and Data Mining in Pattern Recognition* (pp. 50–64).

Boullé, M. (2004). Khiops: A statistical discretization method of continuous attributes. *Machine Learning, 55*(1), 53–69.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. California: Wadsworth International.

Catlett, J. (1991). On changing continuous attributes into ordered discrete attributes. In *Proceedings of the European Working Session on Learning* (pp. 87–102). Springer-Verlag.

Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning*. (pp. 194–202) San Francisco, CA: Morgan Kaufmann.

Elomaa, T., & Rousu, J. (1996). Finding optimal multi-splits for numerical attributes in decision tree learning. *Technical report, NeuroCOLT Technical Report NC-TR-96-041*. Royal Holloway, University of London.

Elomaa, T., & Rousu, J. (1999). General and efficient multisplitting of numerical attributes. *Machine Learning, 36*, 201–244.

Fayyad, U., & Irani, K. (1992). On the handling of continuous-valued attributes in decision tree generation. *Machine Learning, 8*, 87–102.

Fischer, W. D. (1958). On grouping for maximum of homogeneity. *Journal of the American Statistical Association, 53*, 789–798.

Fulton, T., Kasif, S., & Salzberg, S. (1995). Efficient algorithms for finding multi-way splits for decision trees. In *Proceeding of the Twelfth International Conference on Machine Learning*.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning, 11*, 63–90.

Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. *Applied Statistics, 29*(2), 119–127.

Kerber, R. (1991). Chimerge discretization of numeric attributes. In *Proceedings of the 10th International Conference on Artificial Intelligence* (pp. 123–128).

Kohavi, R., & Sahami, M. (1996). Error-based and entropy-based discretization of continuous features. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (pp. 114–119). Menlo Park, CA: AAAI Press/MIT Press.

Kononenko, I., Bratko, I., & Roskar, E. (1984). *Experiments in automatic learning of medical diagnostic rules* (Technical Report). Ljubljana: Joseph Stefan Institute, Faculty of Electrical Engineering and Computer Science.

Lechevallier, Y. (1990). Recherche d'une partition optimale sous contrainte d'ordre total. *Technical report N° 1247*, INRIA.

Liu, H., Hussain, F., Tan, C. L., & Dash, M. (2002). Discretization: An enabling technique. *Data Mining and Knowledge Discovery, 6*(4), 393–423.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica, 14*, 465–471.

Vitanyi, P. M. B., & Li, M. (2000). Minimum description length induction, Bayesianism, and Kolmogorov Complexity. *IEEE Trans. Inform. Theory, IT-46*(2), 446–464.

Zighed, D. A., Rabaseda, S., & Rakotomalala, R. (1998). Fusinter: A method for discretization of continuous attributes for supervised learning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6*(33), 307–326.

Zighed, D. A., Rabaseda, S., Rakotomalala, R., & Feschet F. (1999). Discretization methods in supervised learning. In *Encyclopedia of Computer Science and Technology,* vol. 40 (pp. 35–50) Marcel Dekker Inc.

Zighed, D. A., & Rakotomalala, R. (2000). *Graphes d'induction*. (pp. 327–359) HERMES Science Publications.

B

A Bayes Optimal Approach for Partitioning the Values of Categorical Attributes

Journal of Machine Learning Research 6 (2005) 1431-1452 Submitted 10/04; Revised 3/05; Published 9/05

A Bayes Optimal Approach for Partitioning the Values of Categorical Attributes

Marc Boullé MARC.BOULLE@FRANCETELECOM.COM
France Telecom R&D
2 Avenue Pierre Marzin
22300 Lannion, France

Editor: Greg Ridgeway

Abstract

In supervised machine learning, the partitioning of the values (also called grouping) of a categorical attribute aims at constructing a new synthetic attribute which keeps the information of the initial attribute and reduces the number of its values. In this paper, we propose a new grouping method MODL[1] founded on a Bayesian approach. The method relies on a model space of grouping models and on a prior distribution defined on this model space. This results in an evaluation criterion of grouping, which is minimal for the most probable grouping given the data, *i.e.* the Bayes optimal grouping. We propose new super-linear optimization heuristics that yields near-optimal groupings. Extensive comparative experiments demonstrate that the MODL grouping method builds high quality groupings in terms of predictive quality, robustness and small number of groups.

Keywords: data preparation, grouping, Bayesianism, model selection, classification, naïve Bayes

1 Introduction

Supervized learning consists of predicting the value of a class attribute from a set of explanatory attributes. Many induction algorithms rely on discrete attributes and need to discretize continuous attributes or to group the values of categorical attributes when they are too numerous. While the discretization problem has been studied extensively in the past, the grouping problem has not been explored so deeply in the literature. The grouping problem consists in partitioning the set of values of a categorical attribute into a finite number of groups. For example, most decision trees exploit a grouping method to handle categorical attributes, in order to increase the number of instances in each node of the tree (Zighed and Rakotomalala, 2000). Neural nets are based on continuous attributes and often use a 1-to-N binary encoding to preprocess categorical attributes. When the categories are too numerous, this encoding scheme might be replaced by a grouping method. This problem arises in many other classification algorithms, such as Bayesian networks or logistic regression. Moreover, the grouping is a general-purpose method that is intrinsically useful in the data preparation step of the data mining process (Pyle, 1999).

The grouping methods can be clustered according to the search strategy of the best partition and to the grouping criterion used to evaluate the partitions. The simplest algorithm tries to find the best bipartition with one category against all the others. A more interesting approach consists in searching a bipartition of all categories. The Sequential Forward Selection method derived from that of Cestnik et al. (1987) and evaluated by Berckman (1995) is a greedy algorithm that initializes a group with the best category (against the others), and iteratively adds new categories to this first group. When the class attribute has two values, Breiman et al. (1984) have proposed in CART an optimal method to group the categories into two groups for the Gini criterion. This

[1] This work is covered under French patent number 04 00179.

algorithm first sorts the categories according to the probability of the first class value, and then searches for the best split in this sorted list. This algorithm has a time complexity of O(I log(I)), where I is the number of categories. Based on the ideas presented in (Lechevallier, 1990; Fulton et al., 1995), this result can possibly be extended to find the optimal partition of the categories into K groups in the case of two class values, with the use of a dynamic programming algorithm of time complexity I^2. In the general case of more than two class values, there is no algorithm to find the optimal grouping with K groups, apart from exhaustive search. However, Chou (1991) has proposed an approach based on K-means that allows finding a locally optimal partition of the categories into K groups. Decision tree algorithms often manage the grouping problem with a greedy heuristic based on a bottom-up classification of the categories. The algorithm starts with single category groups and then searches for the best merge between groups. The process is reiterated until no further merge can improve the grouping criterion. The CHAID algorithm (Kass, 1980) uses this greedy approach with a criterion close to ChiMerge (Kerber, 1991). The best merges are searched by minimizing the chi-square criterion applied locally to two categories: they are merged if they are statistically similar. The ID3 algorithm (Quinlan, 1986) uses the information gain criterion to evaluate categorical attributes, without any grouping. This criterion tends to favor attributes with numerous categories and Quinlan (1993) proposed in C4.5 to exploit the gain ratio criterion, by dividing the information gain by the entropy of the categories. The chi-square criterion has also been applied globally on the whole set of categories, with a normalized version of the chi-square value (Ritschard et al., 2001) such as the Cramer's V or the Tschuprow's T, in order to compare two different-size partitions.

In this paper, we present a new grouping method called MODL, which results from a similar approach as that of the MODL discretization method (Boullé, 2004c). This method is founded on a Bayesian approach to find the most probable grouping model given the data. We first define a general family of grouping models, and second propose a prior distribution on this model space. This leads to an evaluation criterion of groupings, whose minimization defines the optimal grouping. We use a greedy bottom-up algorithm to optimize this criterion. The method starts the grouping from the elementary single value groups. It evaluates all merges between groups, selects the best one according to the MODL criterion and iterates this process. As the grouping problem has been turned into a minimization problem, the method automatically stops merging groups as soon as the evaluation of the resulting grouping does not decrease anymore. Additional preprocessing and post-optimization steps are proposed in order to improve the solutions while keeping a super-linear optimization time. Extensive experiments show that the MODL method produces high quality groupings in terms of compactness, robustness and accuracy.

The remainder of the paper is organized as follows. Section 2 describes the MODL method. Section 3 proceeds with an extensive experimental evaluation.

2 The MODL Grouping Method

In this section, we present the MODL approach which results in a Bayes optimal evaluation criterion of groupings and the greedy heuristic used to find a near-optimal grouping.

2.1 Presentation

In order to introduce the issues of grouping, we present in Figure 1 an example based on the Mushroom UCI data set (Blake and Merz, 1998). The class attribute has two values: EDIBLE and POISONOUS. The 10 categorical values of the explanatory attribute CapColor are sorted by decreasing frequency; the proportions of the class values are reported for each explanatory value. Grouping the categorical values does not make sense in the unsupervised context. However, taking the class attribute into account introduces a metric between the categorical values. For example, looking at the proportions of their class values, the YELLOW cap looks closer from the RED cap than from the WHITE cap.

Value	EDIBLE	POISONOUS	Frequency
BROWN	55.2%	44.8%	1610
GRAY	61.2%	38.8%	1458
RED	40.2%	59.8%	1066
YELLOW	38.4%	61.6%	743
WHITE	69.9%	30.1%	711
BUFF	30.3%	69.7%	122
PINK	39.6%	60.4%	101
CINNAMON	71.0%	29.0%	31
GREEN	100.0%	0.0%	13
PURPLE	100.0%	0.0%	10

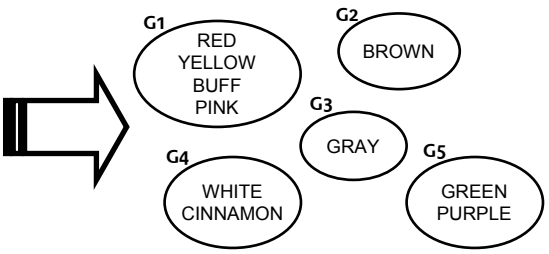


Figure 1. Example of a grouping of the categorical values of the attribute CapColor of data set Mushroom

In data preparation for supervised learning, the problem of grouping is to produce the smallest possible number of groups with the slightest decay of information concerning the class values. In Figure 1, the values of CapColor are partitioned into 5 groups. The BROWN and GRAY caps are kept into 2 separate groups, since their relatively small difference of proportions of the class values is significant (both categorical values have important frequencies). On the opposite, the BUFF cap is merged with the RED, YELLOW and PINK caps: the frequency of the BUFF cap is not sufficient to make a significant difference with the other values of the group.

The issue of a good grouping method is to find a good trade-off between information (as many groups as possible with discriminating proportions of class values) and reliability (the class information learnt on the train data should be a good estimation of the class information observed on the test data). Producing a good grouping is harder with large numbers of values since the risk of overfitting the data increases. In the limit situation where the number of values is the same as the number of instances, overfitting is obviously so important that efficient grouping methods should produce one single group, leading to the elimination of the attribute. In real applications, there are some domains that require grouping of the categorical attributes. In marketing applications for example, attributes such as Country, State, ZipCode, FirstName, ProductID usually hold many different values. Preprocessing these attributes is critical to produce efficient classifiers.

2.2 Definition of a Grouping Model

The objective of the grouping process is to induce a set of groups from the set of values of a categorical explanatory attribute. The data sample consists of a set of instances described by pairs of values: the explanatory value and the class value. The explanatory values are categorical: they can be distinguished from each other, but they cannot *naturally* be sorted. We propose in Definition 1 the following formal definition of a grouping model. Such a model is a pattern that describes both the partition of the categorical values into groups and the proportions of the class values in each group.

Definition 1: A *standard* grouping model is defined by the following properties:

1. the grouping model allows to describe a partition of the categorical values into groups,
2. in each group, the distribution of the class values is defined by the frequencies of the class values in this group.

Such a grouping model is called a SGM model.

Notations:

n: number of instances
J: number of classes
I: number of categorical values
n_i: number of instances for value i
n_{ij}: number of instances for value i and class j
K: number of groups
$k(i)$: index of the group containing value i
n_k: number of instances for group k
n_{kj}: number of instances for group k and class j

The purpose of a grouping model is to describe the distribution of the class attribute conditionally to the explanatory attribute. All the information concerning the explanatory attribute, such as the size of the data set, the number of explanatory values and their distribution might be used by a grouping model. The input data can be summarized knowing n, J, I and n_i. The grouping model has to describe the partition of the explanatory values into groups and the distribution of the class values in each group. A SGM grouping model is completely defined by the parameters $\{ K, \{k(i)\}_{1\le i\le I}, \{n_{kj}\}_{1\le k\le K, 1\le j\le J} \}$.

For example, in Figure 1, the input data consists of n=5865 instances (size of the train data set used in the sample), I=10 categorical values and J=2 class values. The n_i parameters represents the counts of the categorical values (for example, n_1=1610 for the BROWN CapColor).

The grouping model pictured in Figure 1 is defined by K=5 groups, the description of the partition of the 10 values into 5 groups (for example, $k(1)$=2 since the BROWN CapColor belongs to the G2 group) and the description of the distribution of the class values in each group. This last set of parameters ($\{n_{kj}\}_{1\le k\le 5,1\le j\le 2}$) corresponds to the counts in the contingency table of the grouped attribute and class attribute.

2.3 Evaluation of a Grouping Model

In the Bayesian approach, the best model is found by maximizing the probability $P(Model|Data)$ of the model given the data. Using Bayes rule and since the probability $P(Data)$ is constant under varying the model, this is equivalent to maximize $P(Model)P(Data|Model)$. For a detailed presentation on Bayesian theory and its applications to model comparison and hypothesis testing, see for example (Bernardo and Smith, 1994; Kass and Raftery, 1995).

Once a prior distribution of the models is fixed, the Bayesian approach allows to find the optimal model of the data, provided that the calculation of the probabilities $P(Model)$ and $P(Data|Model)$ is feasible. We define below a prior which is essentially a uniform prior at each stage of the hierarchy of the model parameters. We also introduce a strong hypothesis of independence of the distribution of the class values. This hypothesis is often assumed (at least implicitly) by many grouping methods that try to merge similar groups and separate groups with significantly different distributions of class values. This is the case for example with the CHAID grouping method (Kass, 1980), which merges two adjacent groups if their distributions of class values are statistically similar (using the chi-square test of independence).

Definition 2: The following distribution prior on SGM models is called the three-stage prior:

1. the number of groups K is uniformly distributed between 1 and I,
2. for a given number of groups K, every division of the I categorical values into K groups is equiprobable,
3. for a given group, every distribution of class values in the group is equiprobable,
4. the distributions of the class values in each group are independent from each other.

Owing to the definition of the model space and its prior distribution, Bayes formula is applicable to exactly calculate the prior probabilities of the model and the probability of the data given the model. Theorem 1, proven in Appendix A, introduces a Bayes optimal evaluation criterion.

Theorem 1: A SGM model distributed according to the three-stage prior is Bayes optimal for a given set of categorical values if the value of the following criterion is minimal on the set of all SGM models:

$$\log(I)+\log(B(I,K))+\sum_{k=1}^{K}\log\left(C_{n_k+J-1}^{J-1}\right)+\sum_{k=1}^{K}\log\left(n_k!/n_{k,1}!n_{k,2}!...n_{k,J}!\right).$$

C is the combinatorial operator. $B(I,K)$ is the number of divisions of the I values into K groups (with eventually empty groups). When $K=I$, $B(I,K)$ is the Bell number. In the general case, $B(I,K)$ can be written as a sum of Stirling numbers of the second kind $S(I,k)$:

$$B(I,K)=\sum_{k=1}^{K}S(I,k).$$

$S(I,k)$ stands for the number of ways of partitioning a set of I elements into k nonempty sets.

The first term of the criterion in equation 1 stands for the choice of the number of groups, the second term for the choice of the division of the values into groups and the third term for the choice of the class distribution in each group. The last term encodes the probability of the data given the model.

There is a subtlety in the three-stage prior, where choosing a grouping with K groups incorporates the case of potentially empty groups. The partitions are thus constrained to have at most K groups instead of exactly K groups. The intuition behind the choice of this prior is that if K groups are chosen and if the categorical values are dropped in the groups independently from each other, empty groups are likely to appear. We present below two theorems (proven in Appendix A) that bring a more theoretical justification of the choice of the prior. These theorems are no longer true when the prior is to have partition containing exactly K groups.

Definition 3: A categorical value is *pure* if it is associated with a single class.

Theorem 2: In a Bayes optimal SGM model distributed according to the three-stage prior, two pure categorical values having the same class are necessary in the same group.

This brings an intuitive validation of the MODL approach. Furthermore, grouping algorithms can exploit this property in a preprocessing step and considerably reduce their overall computational complexity.

Theorem 3: In a SGM model distributed according to the three-stage prior and in the case of two classes, the Bayes optimal grouping model consists of a single group when each instance has a different categorical value.

This provides another validation of the MODL approach. Building several groups in this case would reflect an over-fitting behavior.

Conjecture 1: In a Bayes optimal SGM model distributed according to the three-stage prior and in the case of two classes, any categorical value whose class proportion is between the class proportions of two categorical values belonging to the same group necessary belongs to this group.

This conjecture has been proven for other grouping criterion such as Gini (Breiman, 1984) or Kolmogorov-Smirnov (Asseraf, 2000) and experimentally validated in extensive experiments for the MODL criterion. It will be considered as true in the following. The grouping algorithms, such as greedy bottom-up merge algorithms, can take benefit from this conjecture. Once the categorical values have been sorted by decreasing proportion of the class, the number of potentially interesting value merges is reduced to (I-1) instead of I(I-1)/2.

2.4 Optimization of a Grouping Model

Once the optimality of an evaluation criterion is established, the problem is to design a search algorithm in order to find a grouping model that minimizes the criterion. In this section, we present a greedy bottom-up merge heuristic enhanced with several preprocessing and post-optimization algorithms whose purpose is to achieve a good trade-off between the time complexity of the search algorithm and the quality of the groupings.

2.4.1 Greedy Bottom-Up Merge Heuristic

In this section, we present a standard greedy bottom-up heuristic. The method starts with initial single value groups and then searches for the best merge between groups. This merge is performed if the MODL evaluation criterion of the grouping decreases after the merge and the process is reiterated until not further merge can decrease the criterion.

The algorithm relies on the O(I) marginal counts, which require one O(n) scan of the data set. However, we express the complexity of the algorithm in terms of n (rather than in terms of I), since the number of categorical values can reach O(n) in the worst case. With a straightforward implementation of the algorithm, the method runs in O(n^3) time (more precisely O(n+I^3)). However, the method can be optimized in O(n^2 *log(n)*) time owing to an algorithm similar to that presented in (Boullé, 2004a). The algorithm is mainly based on the additivity of the evaluation criterion. Once a grouping is evaluated, the value of a new grouping resulting from the merge between two adjacent groups can be evaluated in a single step, without scanning all the other groups. Minimizing the value of the groupings after the merges is the same as maximizing the related variation of value Δvalue. These Δvalues can be kept in memory and sorted in a maintained sorted list (such as an AVL binary search tree for example). After a merge is completed, the Δvalues need to be updated only for the new group and its adjacent groups to prepare the next merge step.

Optimized greedy bottom-up merge algorithm:

- Initialization
  - Create an elementary group for each value: O(n)
  - Compute the value of this initial grouping: O(n)
  - Compute the Δvalues related to all the possible merges of 2 values: O(n^2)
  - Sort the possible merges: O(n^2 log(n))
- Optimization of the grouping: repeat the following steps (at most n steps)
  - Search for the best possible merge: O(1)
  - Merge and continue if the best merge decreases the grouping value
    - Compute the Δvalues of the remaining group merges adjacent to the best merge: O(n)
    - Update the sorted list of merges: O(n log(n))

In the case of two classes, the time complexity of the greedy algorithm can be optimized down to O(n log(n)) owing to conjecture 1.

2.4.2 Preprocessing

In the general case, the computational complexity is not compatible with large real databases, when the categorical values becomes too numerous. In order to keep a super-linear time complexity, the initial categorical values can be preprocessed into a new set of values of cardinality $I' \leq \sqrt{n}$.

A first straightforward preprocessing step is to merge pure values having the same class. This step is compatible with the optimal solution (see Theorem 2).

A second preprocessing step consists in building J groupings for each "one class against the others" sub-problems. This require $O(J\ n\ \log(n))$ time. The subparts of the groups shared by all the J groupings can easily be identified and represent very good candidate subgroups of the global grouping problem. The number of these subparts is usually far below the number of initial categorical values and helps achieving a reduced sized set of preprocessed values.

A third preprocessing step can be applied when the number of remaining preprocessed values is beyond $\sqrt{n}$. The values can be sorted by decreasing frequencies and the exceeding infrequent values can be unconditionally grouped into J groups according to their majority class. This last step is mandatory to control the computational complexity of the algorithm. However, experiments show that this last step is rarely activated in practice.

2.4.3 Post-Optimizations

The greedy heuristic may fall in a local optimum, so that time efficient post-optimizations are potentially useful to improve the quality of the solution. Since the evaluation criterion is Bayes optimal, spending more computation time is meaningful.

A first post-optimization step consists in forcing the merges between groups until a single terminal group is obtained and to keep the best encountered grouping. This helps escaping local optima and requires $O(n\ \log(n))$ computation time. Furthermore, in the case of noisy attribute where the optimal grouping consists of a single group, this heuristic guarantees to find the optimal solution.

A second post-optimization step consists in evaluating every move of a categorical value from one group to another. The best moves are performed as long as they improve the evaluation criterion. This process is similar to the K-means algorithm where each value is attracted by its closest group. It converges very quickly, although this cannot be proved theoretically.

A third post-optimization step is a look-ahead optimization. The best merge between groups is simulated and post-optimized using the second step algorithm. The merge is performed in case of improvement. This algorithm looks similar to the initial greedy merge algorithm, except that it starts from a very good solution and incorporates an enhanced post-optimization. Thus, this additional post-optimization is usually triggered for only one or two extra merges.

3 Experiments

In our experimental study, we compare the MODL grouping method with other supervised grouping algorithms. In this section, we introduce the evaluation protocol, the alternative evaluated grouping methods and the evaluation results on artificial and real data sets. Finally, we present the impact of grouping as a preprocessing step to the Naïve Bayes classifier.

3.1 Presentation

In order to evaluate the intrinsic performance of the grouping methods and eliminate the bias of the choice of a specific induction algorithm, we use a protocol similar as (Boullé, 2004b), where each grouping method is considered as an elementary inductive method which predicts the distribution of the class values in each learned groups.

The grouping problem is a bi-criteria problem that tries to compromise between the predictive quality and the number of groups. The optimal classifier is the Bayes classifier: in the case of an univariate classifier based on a single categorical attribute, the optimal grouping is to do nothing,

i.e. to build one group per categorical value. In the context of data preparation, the objective is to keep most of the class conditional information contained in the attribute while decreasing the number of values. In the experiments, we collect both the number of groups and the predictive quality of the grouping. The number of groups is easy to calculate. The quality of the estimation of the class conditional information hold by each group is more difficult to evaluate.

We choose not to use the accuracy criterion because it focuses only on the majority class value and cannot differentiate correct predictions made with probability 1 from correct predictions made with probability slightly greater than 0.5. Furthermore, many applications, especially in the marketing field, rely on the scoring of the instances and need to evaluate the probability of each class value. In the case of categorical attributes, we have the unique opportunity of observing the class conditional distribution on the test data set: for each categorical value in the test data set, the observed distribution of the class values can be estimated by counting. The grouping methods allow to induce the class conditional distribution from the train data set: for each learnt group on the train data set, the learnt distribution of the class values can be estimated by counting. The objective of grouping is to minimize the distance between the learnt distribution and the observed distribution. This distance can be evaluated owing to a divergence measure, such as the Kullback-Leibler divergence, the chi-square divergence or the Hellinger coefficient. In our experiments, we choose to evaluate this distance using the Kullback-Leibler divergence (Kullback, 1968).

The MODL grouping methods exploits a space of class conditional distribution models (the SGM models) and searches the most probable model given the train data. It is noteworthy that no loss function is optimized in this approach: neither the classification accuracy is optimized nor the Kullback-Leibler divergence (which would require to divide the train data set into train data and validation data). The MODL method exploits all the available train data to build its grouping model.

The evaluation conducted in the experiments focuses on the quality of the groupings (size and Kullback-Leibler divergence criterions) as a preprocessing method for data mining. It is interesting to examine whether optimizing the posterior probability of a grouping model (on the basis on the three-stage prior) leads to high-quality groupings.

3.2 The Evaluation Protocol

The predictive quality of the groupings is evaluated owing to the Kullback-Leibler divergence (Kullback, 1968) applied to compare the distribution of the class values estimated from the train data set with the distribution of the class values observed on the test data set.

Let n' be the number of instances, n'_i be the number of instances for value i and n'_{ij} the number of instances for value i and class j on the test data set.

For a given categorical value i, let p_{ij} be the probability of the j^{th} class value estimated on the train data set (on the basis of the group containing the categorical value), and q_{ij} be the probability of the j^{th} class value observed on the test data set (using directly the categorical value). The q_{ij} probabilities are estimated with the Laplace's estimator in order to deal with zero values. We get

$$p_{ij} = \frac{n_{k(i)j}}{n_{k(i)}} \text{ and } q_{ij} = \frac{n'_{ij}+1}{n'_i+J}.$$

The mean of the Kullback-Leibler divergence on the test data set is

$$D(p||q) = \sum_{i=1}^{I} \frac{n'_i}{n'} \sum_{j=1}^{J} p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

In a first experiment, we compare the behavior of the evaluated grouping method on synthetic data sets, where the ideal grouping pattern is known in advance. In the second experiments, we

use real data sets to compare the grouping methods considered as univariate classifiers. In a third experiment, we use the same data sets to evaluate the results of Naïve Bayes classifiers using the grouping methods to preprocess the categorical attributes. In this last experiment, the results are evaluated using the classification accuracy both on train data sets and on test data sets.

Data Set	Continuous Attributes	Categorical Attributes	Size	Class Values	Majority Accuracy
Adult	7	8	48842	2	76.07
Australian	6	8	690	2	55.51
Breast	10	0	699	2	65.52
Crx	6	9	690	2	55.51
Heart	10	3	270	2	55.56
HorseColic	7	20	368	2	63.04
Ionosphere	34	0	351	2	64.10
Mushroom	0	22	8416	2	53.33
TicTacToe	0	9	958	2	65.34
Vehicle	18	0	846	4	25.77
Waveform	40	0	5000	3	33.84
Wine	13	0	178	3	39.89

Table 1. Data sets

We gathered 12 data sets from U.C. Irvine repository (Blake and Merz, 1998), each data set has at least a few tenths of instances for each class value and some categorical attributes with more than two values. In order to increase the number of categorical attributes candidate for grouping, the continuous attributes have been discretized in a preprocessing step with a 10 equal-width unsupervised discretization. The 12 data sets comprising 230 attributes are described in Table 1; the last column corresponds to the accuracy of the majority class.

The categorical attributes in these data sets hold less than 10 values on average (from an average 3 values in the TicTacToe attributes to about 20 values in the HorseColic attributes). In order to perform more discriminating experiments, we use a second collection of data sets containing all the cross-products of the attributes. In this "bivariate" benchmark, the 12 data sets contain 2614 categorical attributes holding 55 values on average.

3.3 The Evaluated Methods

The grouping methods studied in the comparison are:

- MODL, the method described in this paper,
- Khiops (Boulle, 2004b),
- BIC (Ritschard, 2003),
- CHAID (Kass, 1980),
- Tschuprow (Ritschard et al., 2001),
- Gain Ratio (Quinlan, 1993).

All these methods are based on a greedy bottom-up algorithm that iteratively merges the categories into groups, and automatically determines the number of groups in the final partition of the categories. The MODL method is based on a Bayesian approach and incorporates preprocessing and post-optimization algorithms. The Khiops, CHAID, Tschuprow and BIC methods use the chi-square statistics in different manner. The Gain Ratio method is based on entropy.

The CHAID method is the grouping method used in the CHAID decision tree classifier (Kass, 1980). It applies the chi-square criterion locally to two categorical values in the

contingency table, and iteratively merges the values as long as they are statistically similar. The significance level is set to 0.95 for the chi-square threshold. The Tschuprow method is based on a global evaluation of the contingency table, and uses the Tschuprow's T normalization of the chi-square value to evaluate the partitions. The Khiops method also applies the chi-square criterion on the whole contingency table, but it evaluates the partition using the confidence level related to the chi-square criterion. Furthermore, the Khiops method provides a guaranteed resistance to noise: any categorical attribute independent from the class attribute is grouped in a single terminal group with a user defined probability. This probability is set to 0.95 in the experiments. The BIC method is based on the deviance G^2 statistics, which is a chi-square statistics. It exploits a Bayesian information criterion (Schwarz, 1978) to select the best compromise model between fit and complexity. The Gain Ratio method is the methods used in the C4.5 decision tree classifier (Quinlan, 1993). The gain ratio criterion attempts to find a trade-off between the information on the class values (information gain) and the complexity of the partition (the split information) by dividing the two quantities.

We have re-implemented these alternative grouping approaches in order to eliminate any variance resulting from different cross-validation splits.

3.4 The Artificial Data Sets Experiments

Using artificial data sets allows controlling the distribution of the explanatory values and of the class values. The evaluation of the groupings learned on train data sets can thus be optimal, without requiring any test data set. In the case of grouping, an artificial data set containing one categorical attribute and one class attribute is completely specified by the following parameters:

I: number of categorical values,

J: number of classes,

p_i, $1 \leq i \leq I$: probability distribution of the categorical values,

$p_{j/i}$, $1 \leq j \leq J$, $1 \leq i \leq I$: probability distribution of the class values conditionally to the categorical values.

3.4.1 The Noise Pattern

The purpose of this experiment is to compare the robustness of the grouping methods. The *noise pattern* data set consists of an explanatory categorical attribute independent from the class attribute. The explanatory attribute is uniformly distributed (p_i=1/I) and the class attribute consists of two equidistributed class values ($p_{j/i}$=1/2). We use randomly generated train samples of size 1000 and perform the experiment 1000 times, for different numbers of categorical values. In the case of independence, the optimal number of groups is 1. In Figure 2, we report the mean of the number of unnecessary groups (K-1) and of the Kullback-Leibler divergence between the estimated class distribution and the true class distribution.

The results demonstrate the effectiveness of the Kullback-Leibler divergence to evaluate the quality of a grouping. In the case of attribute independence, the classification accuracy is uniformly equal to 50% whatever the number of groups: it is useless as an evaluation criterion. The most comprehensible grouping consists of a single group whereas the worst one is to build one group per value. The Kullback-Leibler divergence is able to exploit this by a better evaluation of the true class distribution in the most frequent groups. Thus, building separate groups that could be merged leads to a poorer estimation of the class distribution.

The CHAID method creates more and more groups when the number of categorical values increases. This translates by a quickly decreasing quality of estimation of the class distribution, as pictured in Figure 2.

The BIC method performs better than the CHAID method when the number of categorical values is small. When this number of values increases, the BIC and CHAID methods perform similarly.

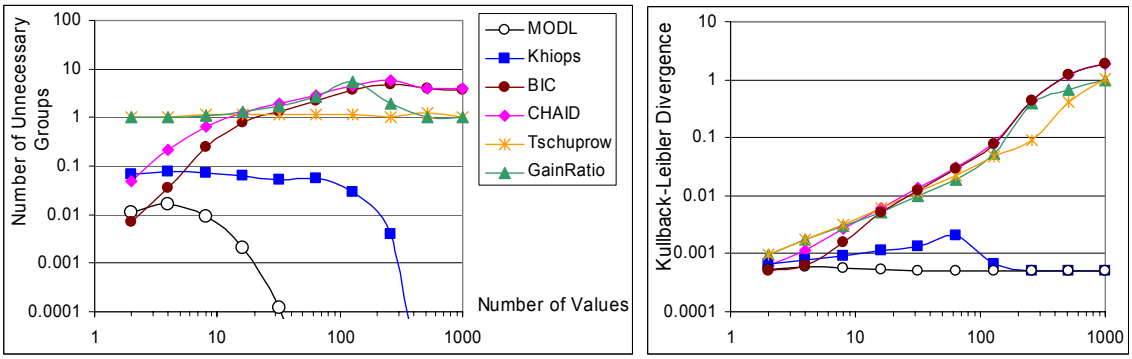


Figure 2. Mean of the number of unnecessary groups (K-1) and of the Kullback-Leibler divergence of the groupings of an explanatory attribute independent from the class attribute

The GainRatio method is constrained to produce at least 2 groups. It overfits the train data as soon as the number of categorical values exceeds a few tens.

The Tschuprow method is also constrained to produce at least 2 groups. It builds almost systematically exactly two groups. A closer look at the Tschuprow criterion shows that the criterion can reach its theoretical bound only when the contingency table is square, meaning that the number of groups is exactly equal to the number of class values. Additional experiments with different numbers of class values (not reported in this paper) confirm this bias. Although the number of groups is constant under varying the number of categorical values, the estimation of the class distribution worsens with higher number of values. This is explained since less frequent values lead to a less reliable estimation of the class probabilities.

The Khiops method is designed to build one single group with probability 0.95, when the explanatory attribute and the class attribute are independent. This is confirmed by the experiment up to about 100 categorical values. Above this number of values, the Khiops method systematically builds one group for the reason that it unconditionally groups the least frequent values in a preprocessing step. The objective of this preprocessing step is to improve the reliability of the confidence level associated with the chi-square criterion, by constraining every cell of the contingency table to have an expected value of at least 5.

The MODL method builds one single group almost always, and the proportion of multi-groups partitions decreases sharply with the number of categorical values. The experiments have been performed 100000 times for the MODL method. Above 50 values, no grouping (out of 100000) contains more than one group.

3.4.2 The Mixture Pattern

The objective of this experiment is to evaluate the sensibility of the grouping methods. The *mixture pattern* data set consists of an explanatory categorical attribute distributed according to several mixtures of class values. The explanatory attribute is uniformly distributed (p_i=1/I) and the class attribute consists of four equidistributed class values ($p_{j/i}$=1/4). We designed 8 artificial groups corresponding to 8 different mixtures of class values. In each group, one of the class values is the majority class (with probability 0.50 or 0.75) and the other three classes are equidistributed, as pictured in Figure 3.

We use randomly generated train samples of size 10000 and perform the experiment 1000 times, for different numbers of categorical values. Due to the quadratic complexity of the algorithms, the experiment was conducted up to 2000 categorical values, except for the MODL algorithm that has a super-linear complexity. In Figure 4, we report the mean of the number of groups and of the Kullback-Leibler divergence between the estimated class distribution and the true class distribution.

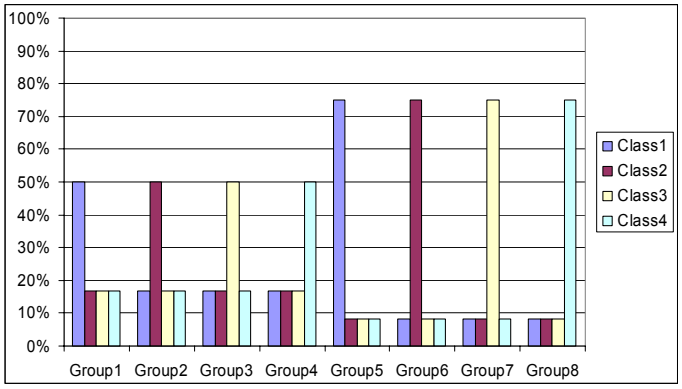


Figure 3. Class distribution in 8 artificial group and 4 class values; the categorical values are uniformly distributed on the 8 groups

The CHAID method overfits the training data by creating too many groups, even more than in the case of independence since the number of class values is now 4. The BIC method manages to find the optimal number of groups when the number of categorical values is below 100. Above this threshold, it overfits the training data and exhibits a behavior similar to that of the CHAID method. The behavior of the GainRatio method is unexpected. For small numbers of categorical values, it builds a constant number of groups equal to the number of class values, and beyond one hundred values, the number of groups raises sharply. The Tschuprow method is so strongly biases that it always produces exactly 4 groups. The Khiops method benefits from its robustness and correctly identifies the 8 artificial groups as long as the frequencies of the categorical values are sufficient. Beyond about 400 values, the minimum frequency constraint of the Khiops algorithms become active and the number of groups falls down to 1. The MODL method almost always builds optimal groupings with the correct number of groups. When the number of categorical values becomes large (about 500, *i.e.* an average frequency of 40 per value), there is a transition in the behavior of the algorithm, that produces only 4 groups instead of 8. The frequency of the categorical values is no longer sufficient to discriminate 8 types of class distributions. When the number of class values increases again (beyond 2000, *i.e.* an average frequency of 5 per value), there is a second transition and the MODL method builds one single group.

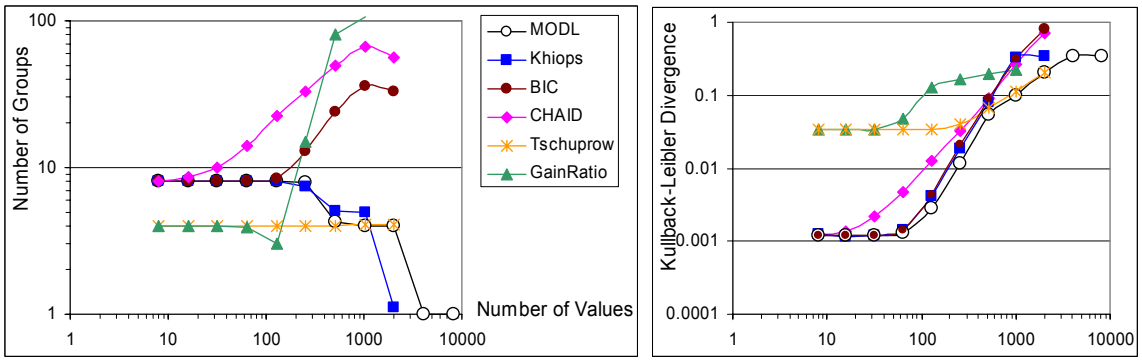


Figure 4. Mean of the number of groups and of the Kullback-Leibler divergence of the groupings of an explanatory attribute distributed in 8 mixtures of class values

To summarize, the noise and mixture pattern experiments are a convenient way to characterize each grouping methods with their bias, robustness, sensibility and limits. The experiments show the interest of using the Kullback-Leibler divergence to evaluate the quality of

the groupings. This criterion looks well suited to evaluate the groupings in real data sets, where the true class distribution is unknown.

The CHAID method exhibits an overfitting behavior, which decays when the number of categorical values or class values increases. The BIC method finds the correct number of groups when the categorical values are not too numerous and then overfits the training data. The Tschuprow is strongly biased in favor of numbers of groups equal to numbers of classes. The GainRatio exhibits a varying biased and overfitting behavior according to the distribution of the train data. The Khiops method is robust but suffers from a lack of sensibility when the categorical values are too numerous, due to its minimum frequency constraint.

The MODL method builds groupings that are optimum, the most probable groupings given the train data. It is the only evaluated method that retrieves the exact number of groups both in case of noise data and of informative data.

3.5 The Real Data Sets Experiments

The goal of this experiment is to evaluate the intrinsic performance of the grouping methods, without the bias of the choice of a specific induction algorithm. The grouping are performed on all the attributes of the UCI data sets presented in Table 1, using a stratified tenfold cross-validation. As the purpose of the experiments is to evaluate the grouping methods according to the way they compromise between the quality and the size of the groupings, we also added three basic grouping methods for comparison reasons. The first method named NoGrouping builds one group per categorical value: it is the least biased method for estimating the distribution of the class values at the expense of the highest number of groups. The second method called ExhaustiveCHAID (SPSS, 2001) is a version of the CHAID method that merges similar pairs continuously until only a single pair remains. We added a similar method ExhaustiveMODL, which allows comparing the two methods when they are constrained to build the same number of groups.

During the experiments, we collect the number of groups and the Kulback-Leibler divergence between the class distribution estimated on train data sets and the class distribution observed on test data sets. For each grouping method, this represents 2300 measures for the univariate analysis (230 attributes) and 26140 measures for the bivariate analysis (2614 pairs of attributes). All these results are summarized across the attributes of the data sets owing to means, in order to provide a gross estimation of the relative performances of the methods. We report the mean of the number of groups and of the Kullback-Leibler divergence for the univariate and bivariate analysis in Figure 5. For the Kullback-Leibler divergence, we use geometric means normalized by the result of the NoGrouping method in order to focus on the ratios of predictive performance between tested methods. The gray line highlights the Pareto curve of the results obtained by the grouping methods.

As expected, the NoGrouping method obtains the best results in term of predictive quality, at the expense of the worst number of groups. However, in the univariate analysis, the Khiops, BIC, CHAID and MODL methods reach almost the same quality with far less groups. The Tschuprow method is hampered by its bias in favor of number of groups equal to the number of class values, so that its performance are not better that those of the constrained methods ExhaustiveCHAID and ExhaustiveMODL. The GainRatio method is dominated by the other methods. The bivariate analysis is much more selective. The results follow the same trend with sharper differences between the methods. The Khiops method underfits the data because of its minimum frequency constraint, while the CHAID method suffers from its lack of overfitting control by producing too many groups and degrading its predictive performance. Although its criterion incorporates a complexity penalty, the BIC method builds too many groups and overfits the data.

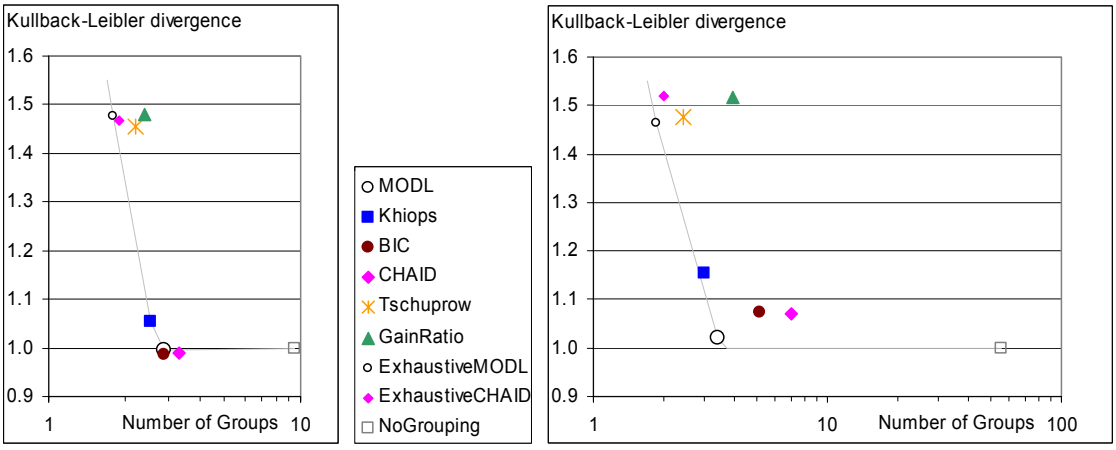


Figure 5. Mean of the number of groups and of the Kullback-Leibler divergence of the groupings performed on the UCI data sets, in univariate analysis (on the left) and bivariate analysis (on the right)

The MODL method gets the lowest number of group without discarding the predictive quality. It manages to reduce the number of categorical values by one order of magnitude while keeping the best estimate of the class conditional probability.

3.6 Impact of Groupings on the Naïve Bayes Classifier

The aim of this experiment is to evaluate the impact of grouping methods on the Naïve Bayes classifier. The Naïve Bayes classifier (Langley et al., 1992) assigns the most probable class value given the explanatory attributes values, assuming conditional independence between the attributes for each class value. The Naïve Bayes classifier is a very simple technique that does not require any parameter setting, which removes the bias due to parameter tuning. It performs surprisingly well on many data sets (Dougherty et al., 1995; Hand and Yu, 2001) and is very robust. High quality grouping tend to increase the frequency in each group and to decrease the variance in the estimation of the conditional class density. It is interesting to examine whether this can benefit to classifiers, even to the Naïve Bayes classifier which is particularly unsophisticated.

The evaluation of probabilities for numeric attributes owing to discretization has already been discussed in the literature (Dougherty et al., 1995; Hsu et al, 2003; Yang and Webb, 2003). Experiments have shown that even a simple Equal Width discretization with 10 bins brings superior performances compared to the assumption using the Gaussian distribution. On the opposite, the probabilities for categorical attribute are estimated using the Laplace's estimator directly on the categorical values, without any preprocessing such as grouping. It is interesting to study whether grouping could produce a more robust estimation of the class distributions and enhance the performance of the Naïve Bayes classifier. The experiment is performed on the 12 data sets presented in Table 1, using the univariate (all categorical attributes) and bivariate (all pairwise interactions of categorical attributes) sets of data sets. A Student's test at the 5% confidence level is performed between the MODL grouping method and the other methods to determine whether the differences of performance are significant. According to (Dietterich, 1998), the McNemar's test is more reliable, but it does not assess the effect of varying the training set. However, these statistical tests "must be viewed as approximate, heuristic tests, rather than as rigorously correct statistical methods" (Dietterich, 1998). Table 2 reports a summary of the test accuracy and robustness results, using the mean of the data set results, the average rank of each method and the number of significant wins and losses of the MODL method.

	Test accuracy						Robustness (Test acc. / Train acc.)					
	Univariate data sets			Bivariate data sets			Univariate data sets			Bivariate data sets		
	Mean	Rank	Wins	Mean	Rank	Wins	Mean	Rank	Wins	Mean	Rank	Wins
MODL	84.8%	2.6		85.8%	2.8		98.6%	3.0		96.8%	4.2	
Khiops	84.1%	3.4	3/1	83.7%	4.7	4/0	98.6%	3.8	1/0	97.2%	2.5	0/3
BIC	83.2%	4.2	4/0	84.9%	3.2	3/0	96.1%	5.2	3/0	93.9%	6.7	7/0
CHAID	83.3%	3.7	2/0	84.7%	4.0	2/1	96.1%	5.2	1/0	93.6%	7.3	7/0
Tschuprow	82.8%	6.4	7/0	83.3%	6.6	6/0	96.6%	5.2	1/0	93.8%	5.5	3/0
GainRatio	81.5%	6.1	5/1	82.9%	5.8	4/0	95.4%	6.0	1/0	93.0%	5.7	5/0
ExMODL	83.4%	5.2	5/0	84.6%	4.7	4/0	98.5%	3.8	0/0	97.1%	2.7	0/2
ExCHAID	81.9%	7.4	7/0	83.2%	5.7	6/0	96.2%	5.8	2/0	94.6%	3.5	2/2
NoGrouping	84.0%	4.1	4/0	84.6%	5.1	4/0	97.3%	6.4	3/0	94.0%	6.2	6/0

Table 2. Summary of the test accuracy and robustness results (mean, average rank and number of wins/losses) of the Naïve Bayes classifier on the UCI data sets, in univariate analysis and bivariate analysis

The results look consistent on the three indicators and show that the MODL method dominates the other methods on the test accuracy criterion. The mean results are pictured in Figure 6 with the classification accuracy reported both on train and test data sets, in order to visualize the train and test accuracy of the methods in a two criteria-analysis. The thick gray line on the diagonal represents the asymptotic best achievable robustness of the methods. The thin gray line highlights the Pareto curve in the two-criteria analysis between robustness and test accuracy.

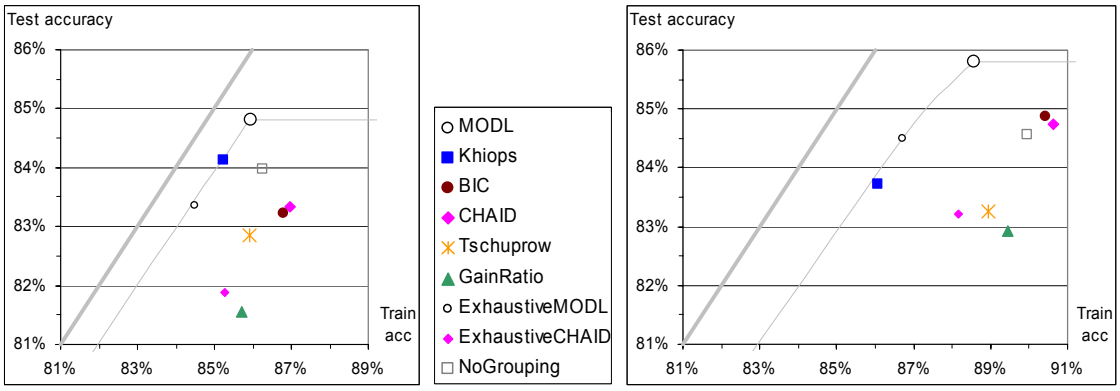


Figure 6. Mean of the train accuracy and test accuracy of the Naïve Bayes classifier on the UCI data sets, in univariate analysis (on the left) and bivariate analysis (on the right)

Most methods do not perform better than the NoGrouping method. This probably explains why the Naïve Bayes classifiers do not make use of groupings in the literature. The MODL method clearly dominates all the other methods, owing to the quality of its groupings. The resulting Naïve Bayes classifier is both the more robust one (together with Khiops) and the more accurate on test data sets. Another important aspect learnt from this experiment is the overall gain in test accuracy when the pairs of attributes are used. The bivariate analysis allows to investigate simple interactions between attributes and to go beyond the limiting independence assumption of the Naïve Bayes classifier. Although this degrades the robustness (because of a decrease in the frequency of the categorical values), this enhances the test accuracy. From univariate to bivariate analysis, the MODL method achieves an increase in test accuracy about twice and a decay in robustness approximately half that of the reference NoGrouping method.

Compared to the NoGrouping method, the MODL method is the only evaluated grouping method that improves the test accuracy of the Naïve Bayes classifier. The most noticeable effect of using the MODL method is a drastic improvement of the robustness.

4 Conclusion

When categorical attributes contain few values, typically less than 10 values, using a grouping method is not required. As the number of values increases (which is common in marketing applications), preprocessing the categorical attributes becomes attractive in order to improve the performance of classifiers. The issue of grouping methods is to reduce the number of groups of values while maintaining the conditional class information.

The MODL grouping method exploits the precise definition of a family of grouping models with a general prior. This provides a new evaluation criterion which is minimal for the Bayes optimal grouping, *i.e.* the most probable grouping given the data sample. An optimization heuristics including preprocessing and post-optimizations is proposed in this paper to optimize the grouping with super-linear time complexity. This algorithm manages to efficiently find high quality groupings.

Extensive evaluations both on real and synthetic data indicate notable performances for the MODL method. It is time efficient and does not require any parameter setting. It builds groupings that are both robust and accurate. The more valuable characteristic of the MODL method is probably the understandability of the groupings. Although understandability is hard to evaluate, the method is theoretically founded to produce correct explanations of the explanatory categorical attributes on the basis of the partition of their values, and even the most probable "grouping based" explanation given the train data.

Acknowledgments

I am grateful to the editor and the anonymous reviewers for their useful comments.

Appendix A

In this appendix, we first present the combinatorial formula used to evaluate the numbers of partition of a set and second provide the proof of the theorems introduced in the paper.

A.1 Partition numbers

The number of ways a set of n elements can be partitioned into nonempty subsets is called a Bell number and denoted B_n.

The number of ways of partitioning a set of n elements into k nonempty subsets is called a Stirling number of the second kind and denoted $S(n,k)$.

The Bell numbers can be defined by the sum

$$B_n = \sum_{k=1}^{n} S(n,k) \ .$$

Let $B(n,k)$ be the number of partition of a set of n elements into at most k parts. This number, that we choose to call a generalized Bell number, can be defined by the sum

$$B(n,k) = \sum_{i=1}^{k} S(n,i) \ .$$

Theorem 1: A SGM model distributed according to the three-stage prior is Bayes optimal for a given set of categorical values if the value of the following criterion is minimal on the set of all SGM models:

$$\log(I)+\log(B(I,K))+\sum_{k=1}^{K}\log\left(C_{n_k+J-1}^{J-1}\right)+\sum_{k=1}^{K}\log\left(n_k!/n_{k,1}!n_{k,2}!..n_{k,J}!\right) .$$

Proof:
The prior probability of a grouping model *M* can be defined by the prior probability of the parameters of the model.

Let us introduce some notations:

- $p(K)$: prior probability of the number of groups *K*,
- $p\left(\{k(i)\}\right)$: prior probability of a partition (defined by $\{k(i)\}$) of the categorical values into *K* groups,
- $p\left(\{n_{kj}\}\right)$: prior probability of the set of parameters $\{n_{11},...,n_{kj},...,n_{KJ}\}$,
- $p\left(\{n_{kj}\}_k\right)$: prior probability of the set of parameters $\{n_{k1},...,n_{kJ}\}$.

The objective is to find the grouping model *M* that maximizes the probability $p(M \mid D)$ for a given train data set *D*. Using Bayes formula and since the probability $p(D)$ is constant under varying the model, this is equivalent to maximize $p(M)\,p(D \mid M)$.

Let us first focus on the prior probability $p(M)$ of the model. We have

$$\begin{aligned} p(M) &= p\left(K,\{k(i)\},\{n_{kj}\}\right) \\ &= p(K)\,p\left(\{k(i)\} \mid K\right)p\left(\{n_{kj}\} \mid K,\{k(i)\}\right). \end{aligned}$$

The first hypothesis of the three-stage prior is that the number of groups is uniformly distributed between 1 and *I*. Thus we get

$$p(K)=\frac{1}{I} .$$

The second hypothesis is that all the partition of the categorical values into at most *K* groups are equiprobable for a given *K*. Computing the probability of one set of groups turns into the combinatorial evaluation of the number of possible group sets. This number is equal to the generalized Bell number $B(I,K)$. Thus we obtain

$$p\left(\{k(i)\} \mid K\right)=\frac{1}{B(I,K)} .$$

The last term to evaluate can be rewritten as a product using the hypothesis of independence of the distributions of the class values between the groups. We have

$$\begin{aligned} p\left(\{n_{kj}\} \mid K,\{k(i)\}\right) &= p\left(\{n_{kj}\}_1,\{n_{kj}\}_2,...,\{n_{kj}\}_K \mid K,\{k(i)\}\right) \\ &= \prod_{k=1}^{K} p\left(\{n_{kj}\}_k \mid K,\{k(i)\}\right) \\ &= \prod_{k=1}^{K} p\left(\{n_{kj}\}_k \mid n_k\right). \end{aligned}$$

The frequencies per group n_k derive from the frequencies per categorical values n_i for a given partition of the values into groups.

For a given group *k* with size n_k, all the distributions of the class values are equiprobable. Computing the probability of one distribution is a combinatorial problem, which solution is

$$p\left(\{n_{kj}\}_k \mid n_k\right)=\frac{1}{C_{n_k+J-1}^{J-1}} .$$

Thus,

$$p\left(\left\{n_{kj}\right\} \mid K, \left\{k(i)\right\}\right) = \prod_{k=1}^{K} \frac{1}{C_{n_k+J-1}^{J-1}} .$$

The prior probability of the model is then

$$p(M) = \frac{1}{I} \frac{1}{B(I,K)} \prod_{k=1}^{K} \frac{1}{C_{n_k+J-1}^{J-1}} .$$

Let us now evaluate the probability of getting the train data set D for a given model M. We first divide the data set D into K subsets D_k of size n_k corresponding to the K groups. Using again the independence assumption between the groups, we obtain

$$\begin{aligned} p(D \mid M) &= p\left(D \mid K, \left\{k(i)\right\}, \left\{n_{kj}\right\}\right) \\ &= p\left(D_1, D_2, \ldots, D_K \mid K, \left\{k(i)\right\}, \left\{n_{kj}\right\}\right) \\ &= \prod_{k=1}^{K} p\left(D_k \mid K, \left\{k(i)\right\}, \left\{n_{kj}\right\}\right) \\ &= \prod_{k=1}^{K} \frac{1}{\left(n_k ! / n_{k,1} ! n_{k,2} ! \ldots n_{k,J} !\right)}, \end{aligned}$$

as evaluating the probability of a subset D_k under uniform prior turns out to be a multinomial problem.

Taking the negative log of the probabilities, the maximization problem turns into the minimization of the claimed criterion

$$\log(I) + \log(B(I,K)) + \sum_{k=1}^{K} \log\left(C_{n_k+J-1}^{J-1}\right) + \sum_{k=1}^{K} \log\left(n_k ! / n_{k,1} ! n_{k,2} ! \ldots n_{k,J} !\right) .$$

Theorem 2: In a Bayes optimal SGM model distributed according to the three-stage prior, two pure categorical values having the same class are necessary in the same group.

Proof:

Let *a* and *b* be the two pure values related to the same class (indexed as class 1 for convenience reasons). Let us assume that, contrary to the claim, the two values are separated into two groups *A1* and *B1*.

We construct two new groups *A0* and *B2* by moving the pure value *a* from *A1* to *B1*. The cost variation $\Delta Cost1$ of the grouping is

$$\begin{aligned} \Delta Cost1 = \Delta PartitionCost1 \\ &+ \log\left((n_{A0} + J - 1)! / n_{A0} !(J-1)!\right) + \log\left(n_{A0} ! / n_{A0,1} ! n_{A0,2} ! \ldots n_{A0,J} !\right) \\ &+ \log\left((n_{B2} + J - 1)! / n_{B2} !(J-1)!\right) + \log\left(n_{B2} ! / n_{B2,1} ! n_{B2,2} ! \ldots n_{B2,J} !\right) \\ &- \log\left((n_{A1} + J - 1)! / n_{A1} !(J-1)!\right) - \log\left(n_{A1} ! / n_{A1,1} ! n_{A1,2} ! \ldots n_{A1,J} !\right) \\ &- \log\left((n_{B1} + J - 1)! / n_{B1} !(J-1)!\right) - \log\left(n_{B1} ! / n_{B1,1} ! n_{B1,2} ! \ldots n_{B1,J} !\right). \end{aligned}$$

If the *A1* group contains only the *a* value, moving *a* from *A1* to *B1* results in a decreased number of groups, with a related variation of partition cost

$$\Delta PartitionCost1 = \log\left(B(I,K-1)\right) - \log\left(B(I,K)\right),$$

which is negative. In the opposite case, the number of groups remains the same and the resulting variation of partition cost is zero.

The frequencies are the same for each class except for class 1, thus

$$\begin{aligned}\Delta Cost1 - \Delta PartitionCost1 &= \log\left((n_{A1} - n_a + J - 1)!/(n_{A1} + J - 1)!\right) \\ &+ \log\left((n_{B1} + n_a + J - 1)!/(n_{B1} + J - 1)!\right) \\ &+ \log\left(n_{A1,1}!/(n_{A1,1} - n_{a,1})!\right) \\ &+ \log\left(n_{B1,1}!/(n_{B1,1} + n_{a,1})!\right), \\ \Delta Cost1 - \Delta PartitionCost1 &= \log\left(\prod_{n=0}^{n_a-1}(n_{A1,1} - n)/(n_{A1} - n + J - 1)\right) \\ &- \log\left(\prod_{n=1}^{n_a}(n_{B1,1} + n)/(n_{B1} + n + J - 1)\right).\end{aligned}$$

Similarly, we construct two groups *A2* and *B0* by moving pure value *b* from *B1* to *A1*. This time, the cost variation $\Delta Cost2$ of the grouping is

$$\begin{aligned}\Delta Cost2 - \Delta PartitionCost2 &= \log\left(\prod_{n=0}^{n_b-1}(n_{B1,1} - n)/(n_{B1} - n + J - 1)\right) \\ &- \log\left(\prod_{n=1}^{n_b}(n_{A1,1} + n)/(n_{A1} + n + J - 1)\right).\end{aligned}$$

Let us assume that

$$n_{A1,1}/(n_{A1} + J - 1) \le n_{B1,1}/(n_{B1} + J - 1).$$

Using the property

$$0 \le z \le x < y \Rightarrow (x - z)/(y - z) \le x/y \le (x + z)/(y + z),$$

we obtain

$$\begin{aligned}\prod_{n=0}^{n_a-1}(n_{A1,1} - n)/(n_{A1} - n + J - 1) &\le (n_{A1,1}/n_{A1} + J - 1)^{n_a} \\ &\le (n_{B1,1}/n_{B1} + J - 1)^{n_a} \le \prod_{n=1}^{n_a}(n_{B1,1} + n)/(n_{B1} + n + J - 1).\end{aligned}$$

Thus, we get $\Delta Cost1 - \Delta PartitionCost1 \le 0$.

On the opposite, let us assume that

$$n_{A1,1}/(n_{A1} + J - 1) \ge n_{B1,1}/(n_{B1} + J - 1).$$

This time, we obtain $\Delta Cost2 - \Delta PartitionCost2 \le 0$.

Since the variations of partition costs are always non-negative, at least one of the two cost variations $\Delta Cost1$ or $\Delta Cost2$ is negative and the initial grouping could not be optimal. As this is contradictory with the initial assumption, the claim follows.

Remark:
If the partition costs are evaluated using the Stirling numbers of the second kind instead of the generalized Bell numbers, this theorem is no longer true, since decreasing the number of groups can result in an increase of the partition cost (for example, $S(I,I\text{-}1) = I(I\text{-}1)/2$ and $S(I,I) = 1$).

Theorem 3: In a SGM model distributed according to the three-stage prior and in the case of two classes, the Bayes optimal grouping model consists of a single group when each instance has a different categorical value.

Proof:
Since each instance has a different categorical value, all the categorical values are pure values associated with one among the J class values. According to Theorem 2, the values having the same class values are necessary in the same group. The optimal grouping contains at most J groups.

Let A and B be two groups and $A \cup B$ the group obtained by merging A and B.

Let n_A, n_B and $n_{A\cup B}$ be the frequencies of these groups, and let $n_{A,j}$, $n_{B,j}$ and $n_{A\cup B,j}$ be the frequencies per class value in these groups.

When the two groups are merged, the number of groups decreases from K to K-1 with the variation of partition cost $\Delta PartitionCost = \log(B(I,K-1)) - \log(B(I,K))$.

The total variation of the grouping cost is

$$\begin{aligned}\Delta Cost &= \Delta PartitionCost \\ &+\left(\log\left(C^{J-1}_{n_{A\cup B}+J-1}\right)+\log\left(n_{A\cup B}!/n_{A\cup B,1}!n_{A\cup B,2}!...n_{A\cup B,J}!\right)\right) \\ &-\left(\log\left(C^{J-1}_{n_A+J-1}\right)+\log\left(n_A!/n_{A,1}!n_{A,2}!...n_{A,J}!\right)\right) \\ &-\left(\log\left(C^{J-1}_{n_B+J-1}\right)+\log\left(n_B!/n_{B,1}!n_{B,2}!...n_{B,J}!\right)\right),\end{aligned}$$

$$\begin{aligned}\Delta Cost &= \Delta PartitionCost \\ &+\log\left((n_{A\cup B}+J-1)!(J-1)!/(n_A+J-1)!(n_B+J-1)!\right) \\ &-\sum_{j=1}^{J}\log\left(C^{n_{A,j}}_{n_{A\cup B,j}}\right).\end{aligned}$$

Since each class is fully contained either in group A or B, we obtain

$$\Delta Cost = \Delta PartitionCost + \log\left((n_{A\cup B}+J-1)!(J-1)!/(n_A+J-1)!(n_B+J-1)!\right).$$

We are in the case of 2 class values and thus have $J = 2$, $n_{A\cup B} = n$, $K = 2$.

$$\begin{aligned}\Delta Cost &= \log\left(B(n,1)\right)-\log\left(B(n,2)\right)+\log\left((n+1)!/(n_A+1)!(n_B+1)!\right) \\ &= -\log\left(2^{n-1}\right)+\log\left(C^{n_A+1}_{n+2}\right)-\log(n+2).\end{aligned}$$

Since we have $C^k_n = C^{k-1}_{n-1} + C^k_{n-1} \le 2^{n-1}$ for $n > 1$ and $k > 1$ (with a strict inequality when $n > 2$), we finally obtain

$$\begin{aligned}\Delta Cost &< -\log\left(2^{n-1}\right)+\log\left(2^{n+1}\right)-\log(n+2) \\ &< \log\left(4/(n+2)\right) \\ &< 0.\end{aligned}$$

The claim follows.

Remark:
If the partition costs are evaluated using the Stirling numbers of the second kind instead of the generalized Bell numbers, this theorem is no longer true. In particular, when the class values are equi-distributed, the grouping cost in the case of a single group ($\log(n)+\log(n+1)+\log\left(C^{n/2}_n\right)$) is higher than the grouping cost in the case of one group per categorical value ($\log(n)+n\log(2)$).

References

M. Asseraf. Metric on decision trees and optimal partition problem. *International Conference on Human System Learning, Proceedings of CAPS'3*, Paris, 2000.

N. C. Berckman. Value grouping for binary decision trees. Technical Report, Computer Science Department – University of Massachusetts, 1995.

J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. Wiley, New York, 1994.

C. L. Blake and C. J. Merz. UCI Repository of machine learning databases Web URL *http://www.ics.uci.edu/~mlearn/MLRepository.html*. Irvine, CA: University of California, Department of Information and Computer Science, 1998.

M. Boullé. Khiops: a Statistical Discretization Method of Continuous Attributes. *Machine Learning*, 55(1):53-69, 2004a.

M. Boullé. A robust method for partitioning the values of categorical attributes. *Revue des Nouvelles Technologies de l'Information, Extraction et gestion des connaissances (EGC'2004)*, RNTI-E-2, volume II: 173-182, 2004b.

M. Boullé. A Bayesian Approach for Supervised Discretization. *Data Mining V*, Eds Zanasi, Ebecken, Brebbia, WIT Press, pp 199-208, 2004c.

L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone. *Classification and Regression Trees*. California: Wadsworth International, 1984.

B. Cestnik, I. Kononenko and I. Bratko. ASSISTANT 86: A knowledge-elicitation tool for sophisticated users. In Bratko and Lavrac (Eds.), *Progress in Machine Learning*. Wilmslow, UK: Sigma Press, 1987.

P. A. Chou. Optimal Partitioning for Classification and Regression Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):340-354, 1991.

T. G. Dietterich. Approximate Statistical Tests for Comparing Supervised Classification Methods. *Neural Computation*, 10(7), 1998.

J. Dougherty, R. Kohavi and M. Sahami. Supervised and Unsupervised Discretization of Continuous Features. *Proceedings of the Twelf International Conference on Machine Learning*. Los Altos, CA: Morgan Kaufmann, pp 194-202, 1995.

T. Fulton, S. Kasif and S. Salzberg. Efficient algorithms for finding multi-way splits for decision trees. In *Proc. Thirteenth International Joint Conference on Artificial Intelligence*, San Francisco, CA: Morgan Kaufmann, pp 244-255, 1995.

D. J. Hand and K. Yu. Idiot Bayes ? not so stupid after all? *International Statistical Review*, 69:385-398, 2001.

C. N. Hsu, H. J. Huang and T. T Wong. Implications of the Dirichlet Assumption for Discretization of Continuous Variables in Naive Bayesian Classifiers. *Machine Learning*, 53(3):235-263, 2003.

G. V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29(2):119-127, 1980.

R. Kass and A. Raftery. Bayes factors. In *Journal of the American Statistical Association*, 90: 773-795, 1995.

R. Kerber. Chimerge discretization of numeric attributes. *Proceedings of the 10th International Conference on Artificial Intelligence*, pp 123-128, 1991.

S. Kullback. *Information Theory and Statistics*. New York: Wiley, (1959); republished by Dover, 1968.

P. Langley, W. Iba and K. Thompson. An analysis of Bayesian classifiers. In *Proceedings of the 10th national conference on Artificial Intelligence*, AAAI Press, pp 223-228, 1992.

Y. Lechevallier. Recherche d'une partition optimale sous contrainte d'ordre total. Technical report N°1247. INRIA, 1990.

D. Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann, 1999.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81-106, 1986.

J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

G. Ritschard, D. A. Zighed and N. Nicoloyannis. Maximisation de l'association par regroupement de lignes ou de colonnes d'un tableau croisé. *Mathématiques et Sciences Humaines*, n°154-155:81-98, 2001.

G. Ritschard. Partition BIC optimale de l'espace des prédicteurs. *Revue des Nouvelles Technologies de l'Information*, 1:99-110, 2003.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461-464, 1978.

SPSS Inc. *AnswerTree 3.0 User's Guide*. Chicago: SPSS Inc, 2001.

Y. Yang and G. Webb. On why discretization works for naïve-Bayes classifiers. *Proceedings of the 16th Australian Joint Conference on Artificial Intelligence (AI)*, 2003.

D. A. Zighed and R. Rakotomalala. *Graphes d'induction*. Hermes Science Publications, pp 327-359, 2000.

C

Optimal Bivariate Evaluation for Supervised Learning using Data Grid Models

Optimal Bivariate Evaluation for Supervised Learning using Data Grid Models

Marc Boullé

France Télécom R&D Lannion,
marc.boulle@orange-ftgroup.com

Abstract. In the domain of data preparation for supervised learning, filter methods for variable ranking are time efficient. However, their intrinsic univariate limitation prevents them from detecting redundancies or constructive interactions between variables. This paper introduces a new method[1] to automatically, rapidly and reliably evaluate the predictive information of a pair of variables. It is based on a partitioning of each input variable, in intervals in the numerical case and in groups of values in the categorical case. The resulting input data grid allows to evaluate the correlation between the two input variables and the output variable. The best joint partitioning is searched owing to a Bayesian model selection approach. Intensive experiments demonstrate the benefits of the approach, especially the significant improvement of accuracy for classification tasks.

1 Introduction

In a data mining project, the data preparation phase aims at constructing a data table for the modeling phase [Pyl99,CCK+00]. The data preparation is both time consuming and critical for the quality of the mining results. It mainly consists in a search of an efficient data representation, based on variable selection. The purpose of variable selection is three-fold: improve the classifier accuracy, reduce the training and deployment time, and ease the comprehensibility of the classifier [GE03,GGHD06]. Two main approaches, filter and wrapper [KJ97], have been studied in the literature. Filter methods evaluate the correlation between the input variables and the output variable, independently of any modeling technique. Wrapper methods search the best subset of variables for a given classification technique, used as a black box. Wrapper methods, which are very time consuming, are restricted to the modeling phase of data mining, as a post-optimization of a classifier. Filter methods are better suited to the data preparation phase, since they do not rely on modeling assumptions. In this paper, we focus on the filter approach.

[1] French patent N° 06 01499

1.1 Univariate filter methods

Univariate filter methods, also called ranking methods, evaluate each input variable individually. They allow to identify informative variables among a very large set of candidate variables. The input variables are ranked according to the method criterion and a subset of the variables can be selected once a threshold is chosen. The simplest way to determine this threshold is to keep as many variables as the modeling technique (often constrained by scalability issues) can handle . Another classical approach is to train the model with several subsets of increasing size. The best subset is chosen according to a tradeoff between the performance of the classifier and the size of the subset.

The most commonly used ranking methods are based on statistical tests [Sap90], such as the chi-square test for categorical input variables, or Student or Fisher-Snedecor tests for numerical input variables. These statistical tests are easy to apply, but they suffer from serious limitations. They are restricted to a Boolean discrimination between dependent and independent variables, which does not provide an accurate ranking of the input variables. They are also subject to strong constraints (minimum expected frequency in each cell of the contingency table for categorical variable, Gaussian distribution for numerical variables). Many alternative measures of associations between two variables have been studied in the context of decision trees [Kas80,BFOS84,Qui93,ZR00]. These criteria are based on a partition of the values of the input variable to evaluate the dependence between the input parts and the output values. Supervised discretization methods split the numerical domain into a set of intervals and supervised value grouping methods partition the input values into groups. Fine grained partitions allow an accurate discrimination of the output values, whereas coarse grain partitions tend to be more reliable. When the size of the partition is a free parameter, the trade-off between information and reliability is an issue. In the MODL approach, supervised discretization [Bou06a] (or value grouping [Bou05]) is considered as a non-parametric model of dependence between the input and output variables. The best partition is found using a Bayesian model selection approach, which provides a measure of association that is both accurate and reliable.

1.2 Multivariate filter methods

Filter methods suffer from their univariate limitation, being unable to reveal interactions between input variables. For example, redundant variables, which bring the same predictive information, cannot be detected. On the opposite, input variables might be uninformative in univariate evaluations and strongly informative in a joint evaluation. These two cases are illustrated in Figure 1, using multiple scatterplots where each point is drawn in a different shape according to its output value. The left diagram shows the case of two redundant variables. The right diagram corresponds to an XOR pattern: each input variable taken alone is a random mixture of the output values, whereas the two variables taken jointly allow a perfect discrimination of the output values.

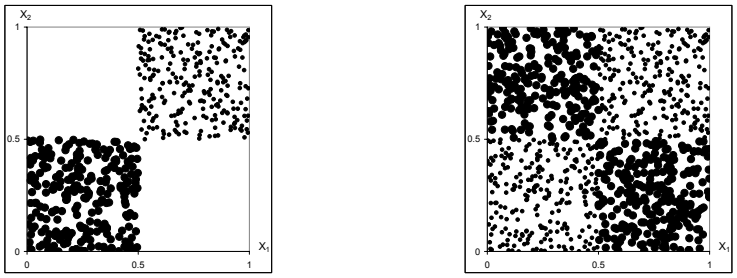


Fig. 1. Multiple scatterplots for two input variables X_1 and X_2, and two output values (small and large circles). The left diagram shows the case of two redundant variables and the right diagram the case of two jointly informative variables

In the case of two numerical input variables, multiple scatterplots are a popular visualization technique to detect interactions between the input and output variables. Scatterplot matrices [CLNL87] extend this technique to sets of input variables and allow to show all pairwise interactions between the variables. These methods are widely used in exploratory data analysis, but they do not provide an evaluation of the joint information contained in the variable pairs. Furthermore, these methods do not apply in the case of large numbers of variables: 100 input variables means about 5000 scatterplots, which cannot be managed by the data analyst.

An automatic evaluation of variable interactions is needed due to the increasing number of variables in datasets. The concept of mutual information between two attributes has been extended to the multivariate case [McG54,Han80] to quantify k-way interactions between variables. The approach is based on comparing the joint information of k variables and that of any subset of at most $(k-1)$ variables. The problem with this approach comes from the evaluation of the joint information, which is the same as evaluating the joint probability distribution function (PDF). In the case of categorical variables, the joint PDF is usually evaluated using the empirical distribution of the data, by counting the number of data instances for each combination of the values. In problems with many variables or with many values per variable, the joint PDF becomes sparse and its empirical evaluation unreliable. The main approach to avoid sparseness of the data is to assume partial independence between variables, which enables to estimate the joint PDF using factorization. The popular naive Bayes classifier [LIT92] relies on the assumption that input variables are independent given the output values. This assumption has been relaxed in semi-naive Bayesian classifiers [Kon91] or in Bayesian network classifiers [FGG97]. Experiments show that these methods improve the naive Bayes accuracy. However, these methods need to analyze dependencies between at least three variables (two input variables and one output variable) and are thus still subject to sparseness of the data. Furthermore, they apply only on categorical data: numerical variables are discretized using a supervised discretization method (MDLPC [FI92] is used in [WBW05] for example), which may destroy the dependencies between input

variables. Another approach to avoid sparseness is to use latent variables. For example in [MC99,VR03], a clustering of the instances is performed for each output value and the discovered clusters are used as latent variables.

1.3 Our contribution

In this paper, we extend the MODL approach to the bivariate case for all pairs of input variables, numerical, categorical or mixed types. Each input variable is partitioned, in intervals in the numerical case and in groups of values in the categorical case. This joint partitioning defines a distribution of the instances in a bi-dimensional input data grid. The correlation between the cells of this data grid and the output values allows to quantify the joint predictive information. The tradeoff between information and reliability is established using a Bayesian model selection approach. This provides an evaluation criterion for any joint partitioning of the input variables. Several optimization heuristics, including pre-optimization and post-optimization are proposed to search the best possible joint-partitioning in a super-linear computation time.

Our method combines several interesting properties. It is able to manage both numerical and categorical input variables. It focuses on the conditional PDF only, is non parametric and non asymptotic. It is regularized to tackle the sparseness issue and optimally strike the balance between informative and reliable models. The models are efficiently optimized in super-linear computation time. Finally, it also provides a filter criterion for the selection of pairs of variables and it builds easily understandable models.

The paper is organized as follows. Section 2 summarizes the MODL method in the univariate case. Section 3 introduces the extension of the approach to the bivariate case and presents the resulting evaluation criterion. Section 4 summarizes the optimization algorithms, which are detailed in [Bou07]. Section 5 evaluates the effectiveness of the method on artificial datasets, where the joint PDF is known in advance. Section 6 demonstrates the benefits of the approach on real datasets, both for the data preparation and data modeling phases of data mining. Finally, section 7 gives a summary.

1.4 Related work

Multivariate discretization or similar techniques have already been proposed in various contexts. For example, the joint partitioning of the lines and rows of contingency table has been studied in the general case [NG05] for data exploration, or in the case of decision trees for the joint partitioning of one input variable and the output variable [ZRES05]. Multivariate discretization has also been developed in the case of association rule mining [Bay01], learning the structure of Bayesian network [SJ04] or for decision rule induction [KK99].

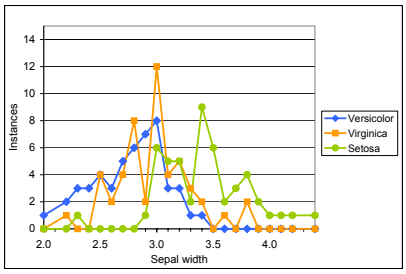
One main difference with these approaches is that our method models the conditional PDF only, not the distribution of the input data. Other major differences are our MAP approach for the evaluation criterion and our optimization algorithm with super-linear computation time.

2 The MODL univariate supervised evaluation methods

This section summarizes the MODL approach for supervised discretization [Bou06a] and value grouping [Bou05].

2.1 The MODL discretization method

The objective of supervised discretization is to induce a list of intervals which splits the numerical domain of a continuous input variable, while keeping the information relative to the output variable. A compromise must be found between information quality (homogeneous intervals in regard to the output variable) and statistical quality (sufficient sample size in every interval to ensure generalization). For example, we present on the left of Figure 2 the number of instances of each class of the Iris dataset [BM96] w.r.t. the sepal width variable. The problem is to find the partition of the numerical domain in intervals which gives us optimal information about the distribution of the data between the three output values.



	]-∞, 2.95[	[2.95, 3.35[	[3.35, +∞[
Versicolor	34	15	1
Virginica	21	24	5
Setosa	2	18	30
Total	57	57	36

Fig. 2. MODL discretization of the Sepal Width variable for the classification of the Iris dataset in 3 classes

In the MODL approach [Bou06a], the discretization is turned into a model selection problem. First, a space of discretization models is defined. The parameters of a specific discretization are the number of intervals, the bounds of the intervals and the output frequencies in each interval. Then, a prior distribution is proposed on this model space. This prior exploits the hierarchy of the parameters: the number of intervals is first chosen, then the bounds of the intervals and finally the output frequencies. The choice is uniform at each stage of the hierarchy. Finally, we assume that the multinomial distributions of the output values in each interval are independent from each other. A Bayesian approach is applied to select the best discretization model, which is found by maximizing the probability $p(Model|Data)$ of the model given the data. Using the Bayes rule and since the probability $p(Data)$ is constant under varying the model, this is equivalent to maximizing $p(Model)p(Data|Model)$.

Let N be the number of instances, J the number of output values, I the number of intervals for the input domain. $N_{i.}$ denotes the number of instances

in the interval i, and N_{ij} the number of instances of output value j in the interval i. In the context of supervised classification, the number of instances N and the number of classes J are supposed to be known. A discretization model is then defined by the parameter set $\left\{I, \{N_{i.}\}_{1\leq i\leq I}, \{N_{ij}\}_{1\leq i\leq I, 1\leq j\leq J}\right\}$.

It is noteworthy that the data partition obtained by applying such a discretization model is invariant by any monotonous variable transformation since it only depends on the variable ranks.

Owing to the definition of the model space and its prior distribution, the Bayes formula is applicable to exactly calculate the prior probabilities of the models and the probability of the data given a model. Taking the negative log of the probabilities, this provides the evaluation criterion given in formula 1.

$$\log N + \log \binom{N+I-1}{I-1} + \sum_{i=1}^{I} \log \binom{N_{i.}+J-1}{J-1} + \sum_{i=1}^{I} \log \frac{N_{i.}!}{N_{i1}!N_{i2}!\ldots N_{iJ}!} \quad (1)$$

The first term of the criterion stands for the choice of the number of intervals and the second term for the choice of the bounds of the intervals. The third term corresponds to the choice of the output distribution in each interval and the last term represents the conditional likelihood of the data given the model. Therefore "complex" models with large numbers of intervals are penalized.

Once the optimality of the evaluation criterion is established, the problem is to design a search algorithm in order to find a discretization model that minimizes the criterion. In [Bou06a], a standard greedy bottom-up heuristic is used to find a good discretization. In order to further improve the quality of the solution, the MODL algorithm performs post-optimizations based on hill-climbing search in the neighborhood of a discretization. The neighbors of a discretization are defined with combinations of interval splits and interval merges. Overall, the time complexity of the algorithm is $O(JN \log N)$.

The MODL discretization method for classification provides the most probable discretization given the data sample. Extensive comparative experiments report high quality performance. In the Iris example, the three intervals of the MODL discretization are shown on the left of Figure 2. The contingency table on the right gives us comprehensible rules such as "for a sepal width less than 2.95, the probability of occurrence of the Versicolor class is $34/57 = 0.60$".

2.2 The MODL value grouping method

Categorical variables are analyzed in a way similar to that of numerical variables, owing to a partitioning model of the input values. In the numerical case, the input values are constrained to be adjacent and the only considered partitions are the partitions in intervals. In the categorical case, there are no such constraints between the values and any partition in groups of values is possible. For instance, Figure 3 illustrates the grouping of the values of the Cap Color variable of the Mushroom dataset [BM96]. The initial input values provide a fine grain

estimation of the class conditional probabilities. The problem is to improve the reliability of this estimation owing to a reduced number of groups of values, while keeping the groups as much informative as possible. Producing a good grouping is harder with large numbers of input values since the risk of overfitting the data increases. In the extreme situation where the number of values is the same as the number of instances, overfitting is obviously so important that efficient grouping methods should produce one single group, leading to the elimination of the variable.

Value	edible	poisonous	Frequency
BROWN	55.2%	44.8%	1610
GRAY	61.2%	38.8%	1458
RED	40.2%	59.8%	1066
YELLOW	38.4%	61.6%	743
WHITE	69.9%	30.1%	711
BUFF	30.3%	69.7%	122
PINK	39.6%	60.4%	101
CINNAMON	71.0%	29.0%	31
GREEN	100.0%	0.0%	13
PURPLE	100.0%	0.0%	10

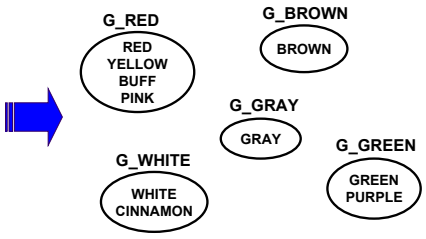


Fig. 3. MODL grouping of the values of the Cap Color variable for the classification of the Mushroom dataset in 2 classes

Let N be the number of instances, V the number of input values, J the number of output values and I the number of groups. $N_{i.}$ denotes the number of instances in the group i, and N_{ij} the number of instances of output value j in the group i. The Bayesian model selection approach is applied like in the discretization case and provides the evaluation criterion given in formula 2. This formula has a similar structure as that of formula 1. The two first terms correspond to the prior distribution of the partitions of the input values, in groups of values in formula 2 and in intervals in formula 1. The two last terms are the same in both formula.

$$\log V + \log B(V,I) + \sum_{i=1}^{I} \log \binom{N_{i.} + J - 1}{J - 1} + \sum_{i=1}^{I} \log \frac{N_{i.}!}{N_{i1}! N_{i2}! \ldots N_{iJ}!} \qquad (2)$$

$B(V,I)$ is the number of divisions of the V values into I groups (with eventually empty groups). When $K = I$, $B(V,I)$ is the Bell number. In the general case, $B(V,I)$ can be written as a sum of Stirling numbers of the second kind.

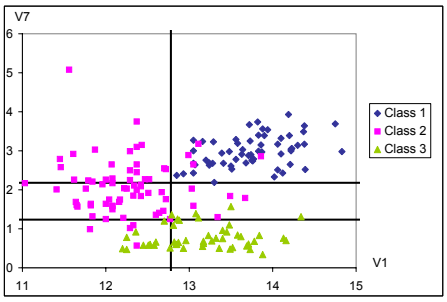
In [Bou05], a standard greedy bottom-up heuristic is proposed to find a good grouping of the input values. Several pre-optimization and post-optimization steps are incorporated, in order to both ensure an algorithmic time complexity of $O(JN\log(N))$ and obtain accurate value groupings.

3 Extension to supervised bivariate evaluation

In this section, we extend the MODL methods to the supervised evaluation of pairs of input variables. We first introduce the approach using an illustrative example and then present the bivariate evaluation criterion in the case of two numerical variables. We generalize the criterion to any case of pairs of input variables and finally introduce the compression gain, a normalized version of the criteria.

3.1 Interest of the joint partitioning of two input variables

Figure 4 draws the multiple scatter plot (per class value) of the input variables V1 and V7 of the Wine dataset [BM96]. This diagram allows to visualize the conditional probability of the output values given the pair of input variables. The V1 variable taken alone cannot separate Class 1 from Class 3 for input values greater than 13. Similarly, the V7 variable is a mixture of Class 1 and Class 2 for input values greater than 2. Taken jointly, the two input variables allow a better separation of the class values.



]2.18, +∞[	(0, 23, 0)	(59, 0, 4)
]1.235, 2.18]	(0, 35, 0)	(0, 5, 6)
]-∞, 1.235]	(0, 4, 11)	(0, 0, 31)
V7xV1	]-∞, 12.78]	]12.78, +∞[

Fig. 4. Multiple scatterplot (per class value) of the input variables V1 and V7 of the Wine dataset. The optimal MODL supervised bivariate partition of the input variables is drawn on the multiple scatterplot, and the triplet of class frequencies per data grid cell is reported in the right table

Extending the univariate case, we partition the dataset on the cross-product of the input variables to quantify the relationship between the input and output variables. Each input variable is partitioned into a set of *parts* (intervals in the numerical case and groups of values in the categorical case). The cross-product of the univariate input partitions defines a *data grid*, which partitions the instances into a set of *data cells*. Each data cell is defined by a pair of parts. The connection between the input variables and the output variable is evaluated owing to the distribution of the output values in each cell of the data grid. It is noteworthy that the considered partitions can be factorized on the input variables.

For instance in Figure 4, the V1 variable is discretized into 2 intervals (one bound 12.78) and the V7 variable into 3 intervals (two bounds 1.235 and 2.18).

The instances of the dataset are distributed in the resulting bidimensional data grid. In each cell of the grid, the distribution of the output values can be estimated by counting. For example, the right table in Figure 4 shows that the cell defined by the intervals $]12.78, +\infty[$ on V1 and $]2.18, +\infty[$ on V7 contains 63 instances. These 63 instances are distributed on 59 instances for Class 1 and 4 instances for Class 3.

Coarse grain data grids tend to be reliable, whereas fine grain data grids allow a better separation of the output values. In our example, the MODL optimal data grid is drawn on the multiple scatter plot on the left of Figure 4.

3.2 Evaluation criterion for pairs of numerical variables

We extend the MODL approach to find the best tradeoff between information and reliability. We introduce in Definition 1 a family of bivariate partitioning models and select the best model owing to a Bayesian model selection approach. We first focus on numerical input variables, before generalizing to any pair of input variables, categorical or mixed types.

Definition 1. *A data grid model is a bivariate partitioning model defined by a partition of each input variable in a set of intervals and by a multinomial distribution of the output values in each cell of the data grid resulting from the cross-product of the univariate partitions.*

Notations.

- Y: output variable,
- X_1, X_2: input variables,
- N: number of instances,
- J: number of output values,
- I_1, I_2: number of intervals for each input variable,
- $N_{i_1..}$: number of instances in the interval i_1 of variable X_1,
- $N_{.i_2.}$: number of instances in the interval i_2 of variable X_2,
- $N_{i_1 i_2.}$: number of instances in the input data cell (i_1, i_2),
- $N_{i_1 i_2 j}$: number of instances of output value j in the input data cell (i_1, i_2).

A data grid model describes the distribution of the output values given the input values. It is completely defined by the numbers of intervals I_1 and I_2, the bounds of the intervals $\{N_{i_1..}\}$ and $\{N_{.i_2.}\}$ and the distribution of the output values $\{N_{i_1 i_2 j}\}$ in each cell (i_1, i_2) of the data grid . It is noteworthy that the numbers of instances per cell $\{N_{i_1 i_2.}\}$ do not belong to the parameters of the data grid models: they are derived from the definition of the two univariate partitions and from the dataset.

Any input information is used to define the family of the model. The bounds of the univariate partition come from the input values and the frequencies of the input data cells come from the dataset. In that sense, the data grid models are data dependent. What is described in the model is the connection between the input variables and the output variable.

We now introduce in Definition 2 a prior distribution on the parameters of the data grid models. This prior exploits the hierarchy of the parameters and is uniform at each stage of this hierarchy.

Definition 2. *The hierarchical prior of the data grid models is defined as follows:*

- *the numbers of input intervals are independent from each other, and uniformly distributed between* 1 *and* N*,*
- *for each input variable and for a given number of intervals, every partition in intervals is equiprobable,*
- *for each cell of the data grid, every distribution of the output values is equiprobable,*
- *the distributions of the output values in each cell are independent from each other.*

We apply the Bayesian model selection approach and obtain the evaluation criterion of a data grid model M in formula 3.

$$\begin{aligned} c(M) &= \log N + \log \binom{N + I_1 - 1}{I_1 - 1} + \log N + \log \binom{N + I_2 - 1}{I_2 - 1} \\ &+ \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \log \binom{N_{i_1 i_2 .} + J - 1}{J - 1} + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \log \frac{N_{i_1 i_2 .}!}{N_{i_1 i_2 1}! N_{i_1 i_2 2}! \ldots N_{i_1 i_2 J}!} \end{aligned} \tag{3}$$

As in the case of univariate discretization (formula 1), the two first terms correspond to the prior probability of the parameters (number of intervals and choice of the bounds) of the discretization of the input variable X_1. Similarly, the two following terms correspond to the prior probability of the discretization of the input variable X_2. The binomial term in the first double sum represents the choice of the multinomial distribution of the output values in each cell. The multinomial term in the last double sum represents the conditional likelihood of the output values given the data grid model.

3.3 Evaluation criterion for any pair of variable

In the case of two categorical variables X_1 and X_2 with V_1 and V_2 input values, we apply the same approach. The X_1 variable is partitioned into I_1 groups of values (instead of intervals in the numerical case) and the X_2 variable into I_2 groups. The distribution of the output values is described in each cell of the data grid resulting from the joint partitioning of the input variables. Compared to the numerical case, the only change is the prior distribution of each univariate partition. The impact in formula 3 is to replace the terms related to the prior distribution of the partition into intervals (two first terms of the univariate discretization of formula 1) by the corresponding value grouping terms (two first terms of the univariate value grouping of formula 2). The resulting evaluation criterion of a data grid model M for two categorical input variables is given in formula 4.

$$c(M) = \log V_1 + \log B(V_1, I_1) + \log V_2 + \log B(V_2, I_2) + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \log \binom{N_{i_1 i_2 .} + J - 1}{J - 1} + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \log \frac{N_{i_1 i_2 .}!}{N_{i_1 i_2 1}! N_{i_1 i_2 2}! \ldots N_{i_1 i_2 J}!} \quad (4)$$

In the mixed case of one categorical variable X_1 with V_1 values and one numerical variable X_2, the first variable is grouped and the second one is discretized. The resulting evaluation criterion of a data grid model M for mixed type input variables is given in formula 5.

$$c(M) = \log V_1 + \log B(V_1, I_1) + \log N + \log \binom{N + I_2 - 1}{I_2 - 1} + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \log \binom{N_{i_1 i_2 .} + J - 1}{J - 1} + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \log \frac{N_{i_1 i_2 .}!}{N_{i_1 i_2 1}! N_{i_1 i_2 2}! \ldots N_{i_1 i_2 J}!} \quad (5)$$

3.4 Compression gain

The evaluation criterion $c(M)$ given in formulas 3, 4, 5 is related to the probability that a data grid model M explains the output variable. The criterion $c(M)$ can also be interpreted as the ability of a data grid model to encode the output values given the input values, since negative log of probabilities are no other than coding lengths [Sha48]. Let $M_\emptyset$ be the null model with only one part for each univariate partition and one cell in the data grid. $c(M_\emptyset)$ represents the coding length of the output values when no input information is used, and is asymptotically equal to N times the Shannon's entropy of the output variable. More complex data grid models may better compress the output values, since the entropy of the output values is defined locally to each input cell. Fine grain cells allow to identify input regions where the output entropy is low (unbalanced mixture of the output values), but too complex data grid models with many cells are penalized with an increasing coding length of the model parameters.

Given these probabilistic and compression interpretations, we propose to use the evaluation criterion $c(M)$ to build a relevance criterion for each pair of input variables. The variable pairs can be sorted by decreasing probability of explaining the output variable. In order to provide a normalized indicator, we consider the following transformation of $c(M)$:

$$g(M) = 1 - \frac{c(M)}{c(M_\emptyset)}. \quad (6)$$

The compression gain $g(M)$ holds its values between 0 and 1 for models which are better than the null model ($g(M)$ is negative otherwise). It has value 0 for the null model and is maximal when the best possible explanation of the output values conditionally to the pair of input variables is achieved.

4 Optimization algorithms

The space of data grid models is so large that straightforward algorithms almost surely fail to obtain good solutions within a practicable computational time. Given that the MODL criterion is optimal, the design of sophisticated optimization algorithms is both necessary and meaningful. Such algorithms are described in [Bou07]. They finely exploit the sparseness of the data grids and the additivity of the MODL criterion, and allow a deep search in the space of data grid models with $O(N)$ memory complexity and a $O(N \log N)$ time complexity.

In this section, we give an overview of the data grid optimization algorithms which are fully detailed in [Bou07].

Let us first focus on the case of two numerical input variables. The optimization of a data grid is a combinatorial problem. For each input variable X_1 and X_2, there are 2^N possible univariate discretizations, which represents $\left(2^N\right)^2$ possible bivariate discretizations. An exhaustive search through the whole space of models is unrealistic. We describe in algorithm 1 a greedy bottom up merge heuristic (GBUM) to optimize the data grids. The method starts with the maximum data grid M_{Max}, which corresponds to the finest possible univariate discretizations, with single value intervals. It evaluates all the merges between adjacent intervals, and performs the best merge if the evaluation criterion decreases after the merge. The process is reiterated until no further merge decreases the criterion.

Algorithm 1 Greedy Bottom Up Merge heuristic (GBUM)

Require: M {Initial data grid solution}
Ensure: $M^*, c(M^*) \leq c(M)$ {Final solution with improved cost}
1: $M^* \leftarrow M$
2: **while** improved solution **do**
3: **for all** Merge m between two parts of variable X_1 or X_2 **do**
4: $M' \leftarrow M^* + m$ {Evaluate merge m on data grid M^*}
5: **if** $c(M') < c(M^*)$ **then**
6: $M^* \leftarrow M'$
7: **end if**
8: **end for**
9: **end while**

Each evaluation of a data grid requires $O(N^2)$ time, since the initial data grid model M_{Max} contains N^2 cells. Each step of the algorithm relies on $O(N)$ evaluations of interval merges, and there are at most $O(N)$ steps, since the data grid becomes equal to the null model $M_\emptyset$ once all the possible merges have been performed. Overall, the time complexity of the algorithm is $O(N^4)$ using a straightforward implementation of the algorithm. However, the method can be optimized in $O(N \log N)$ time, as demonstrated in [Bou07]. The optimized algorithm mainly exploits the sparseness of the data and the additivity of the

evaluation criterion. Although a data grid may contain $O(N^2)$ cells, at most N cells are non empty. Thus, each evaluation of a data grid can be performed in $O(N)$ owing to a specific algorithmic data structure. The additivity of the evaluation criterion means that the criterion can be decomposed on the hierarchy of the components of the data grid: variables, parts and cells. Using this additivity property, all the merges between adjacent parts can be evaluated in $O(N)$ time. Furthermore, when the best merge is performed, the only impacted merges that need to be reevaluated for the next optimization step are the merges that share instances with the best merge. Since the data grid is sparse, the number of reevaluations of data grids is small on average. Sophisticated algorithmic data structures and algorithms are necessary to exploit these optimization principles and guarantee a time complexity of $O(N \log N)$.

The optimized version of the greedy heuristic is time efficient, but it may fall into a local optimum. First, the greedy heuristic may stop too soon and produce too many parts for each input variable. Second, the boundaries of the intervals may be sub-optimal since the merge decisions of the greedy heuristic are never rejected. The post-optimization algorithms described in [Bou06a] in the case of univariate discretization are applied alternatively to each input variable, for a frozen partition of the other input variable.

While post-optimizations may help to refine a good solution, the main heuristic may be unable to obtain such an initial good solution. This problem is tackled using the VNS meta-heuristic [HM01], which mainly benefits from multiple runs of the algorithms with different random initial solutions.

In the case of categorical variables, the combinatorial problem is still worse for large numbers of values V. The number of possible partitions of the values is equal to the Bell number $B(V) = \frac{1}{e} \sum_{k=1}^{\infty} \frac{k^V}{k!}$ which is far greater than the $O(N^2)$ possible discretizations. Furthermore, the number of possible merges between adjacent parts is $O(V^2)$ for categorical variables instead of $O(N)$ for numerical variables. Specific pre-processing and post-processing heuristics are necessary to efficiently handle the categorical input variables. Mainly, the number of groups of values is bounded by $O(\sqrt{N})$ in the algorithms, and the initial and final groupings are locally improved by exchange of values between groups. This allows to keep an $O(N)$ memory complexity and bound the time complexity by $O(N\sqrt{N} \log N)$ for categorical variables.

5 Evaluation on artificial datasets

In this section, the bivariate analysis method is evaluated using the optimization algorithms described in [Bou07]. The evaluation is performed on artificial datasets, where the true data distribution is known. Three patterns are considered: noise, chessboard and Gaussian mixture. An empirical study of the time complexity of the optimization algorithms is also reported.

5.1 Noise pattern

The purpose of the noise pattern experiment is to evaluate the noise resistance of the method, under variation of the sample size. The noise pattern dataset consists of an output variable independent from the input variables. The expected data grid contains one single cell, meaning that the class conditional information is independent from the input variables.

The output variable is equidistributed on two values. In the case of numerical input variables, the input values are uniformly distributed on the [0, 1] numerical domain. In the case of categorical input variables, the input values are equidistributed on V input values. Six families of noise datasets are considered: one for numerical input variables and five for categorical input variables with 2, 8, 32, 128 and 512 values. The experiment is performed on a large range of sample sizes ranging from 2 to 100,000 instances, in a geometric progression. Small sample sizes allow to study non-asymptotic behavior whereas large sample sizes focus on scalability issues.

The evaluated criterion is the number of cells in the data grid. In order to obtain reliable results, the experiment is performed on 100 randomly generated train datasets for each sample size. Figure 5 presents the mean cell number, for each dataset family and each sample size.

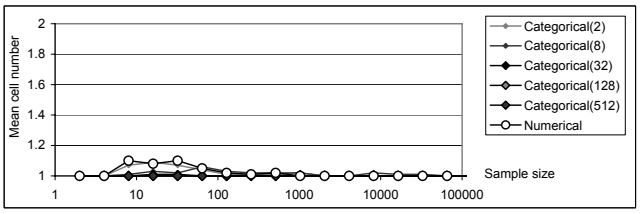


Fig. 5. Mean cell number for the noise pattern datasets, for sample size ranging from 2 to 100,000

Below sample size 4, there are not enough instances to constitute any pattern. Between sample size 10 and 100, a pattern is detected in about 10% of the cases for numerical input variables and for categorical input variables having less than two values. For larger sample sizes, or larger numbers of categorical input values, the noise pattern is almost always correctly detected.

5.2 Chessboard pattern

The purpose of the chessboard pattern experiment is to evaluate the ability of the method to identify complex bivariate patterns, that cannot be detected using univariate analysis. A chessboard pattern of size 2 corresponds to an XOR pattern, as pictured in the right scatterplot of Figure 1. We generalize this XOR

pattern using chessboards having S columns and S rows. Each input value is given an index between 1 and S, and the output value results from the parity of the sum of the indexes of the two input values.

In case of numerical input variables, the input values are uniformly distributed on the [0, S] numerical domain and rounded to the closest ceiling integer to obtain the input indexes. In case of categorical variables, S input values are considered, which directly provide the input indexes. Five sizes of chessboard patterns are considered (2x2, 8x8, 32x32, 128x128, 512x512) both for numerical or categorical input variables. The evaluation protocol is the same as for the noise pattern.

Figure 6 presents the mean cell number for each dataset in the numerical family and for each sample size. The expected data grid contains S^2 cells, resulting from two univariate discretizations of S intervals. The results exhibit the same behavior for all sizes of numerical chessboards. Below a given threshold, the number of instances is not sufficient to detect the pattern, and beyond this threshold, the expected number of cells is rapidly and accurately detected. About 16 instances are necessary to detect the 2x2 pattern, 250 instances for the 8x8 pattern, 2000 instances for the 32x32 pattern, 16000 for the 128x128 pattern and 150000 for the 512x512 pattern. Interestingly, large sample sizes allow to detect very complex patterns with a remarkably small average number of instances per data grid cell. About 4 instances per cell are necessary to detect the 2x2 pattern, and only 0.5 instance per cell on average for the 512x512 pattern.

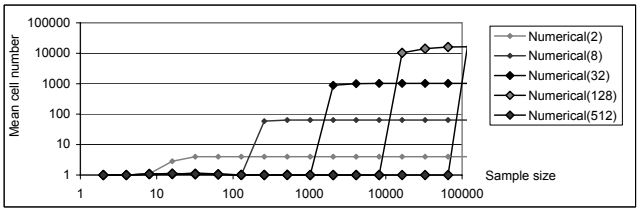


Fig. 6. Mean cell number for the numerical chessboard pattern datasets, for sample size ranging from 2 to 100,000

Figure 7 presents the mean cell number for each dataset in the categorical family and for each sample size. The expected data grid reduces to an XOR pattern with 4 cells, resulting from two univariate groupings of the S values into two groups (according the the parity of the value indexes). Compared to the numerical case, the univariate partitions are more complex to describe, but the output distribution can be described with only 4 cells (instead of S^2). The results exhibit the same kind of behavior as in the numerical case, but the pattern detection thresholds are much smaller. Only 500 instances are sufficient to detect 128x128 pattern and 4000 for the 512x512 pattern, which is remarkably small.

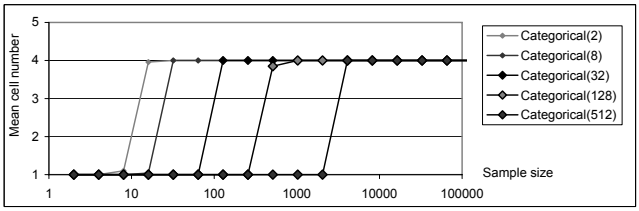


Fig. 7. Mean cell number for the categorical chessboard pattern datasets, for sample size ranging from 2 to 100,000

5.3 Gaussian mixture pattern

The purpose of the Gaussian mixture pattern is to evaluate the limits of the method, when applied to a dataset which is far from its learning bias. Although data grid models are non parametric and able to approximate any distribution provided that there are enough instances, they are biased in favor of constant conditional probabilities in each cell and of partition boundaries which are parallel to the axis. The true distribution in a Gaussian mixture pattern does obviously not fit this bias.

The pattern in the experiment contains two equidistributed output values. For each output value, the input values are distributed according to a bidimensional Gaussian vector with independent variables, as shown in Figure 8.

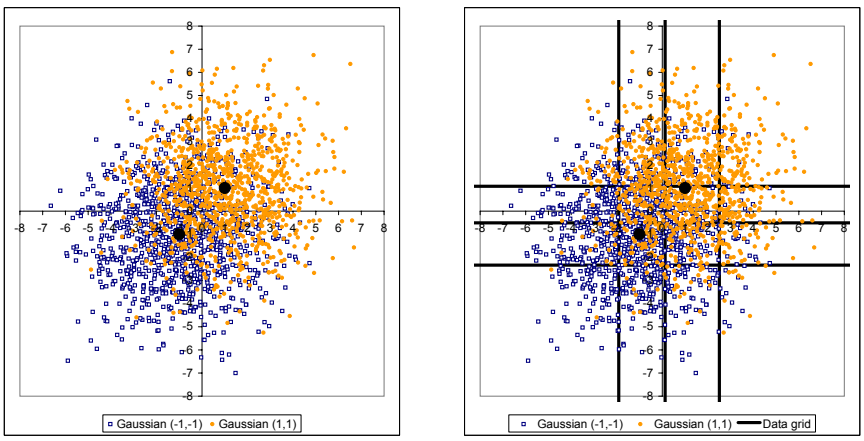


Fig. 8. Gaussian mixture for two Gaussian distributions centered on $(-1,-1)$ and $(+1,+1)$ with standard deviation of 2. The scatterplot displays 2000 instances, and the optimal data grid which contains 16 cells is presented on the right

The evaluation protocol is similar to that of the noise pattern. We also evaluate the root mean square error (RMSE) in order to compare the true conditional probability p, known from the Gaussian mixture parameters, to the estimated conditional probability q, computed from the trained data grid. In each non empty cell (i_1, i_2) of the trained data grid, the conditional probability q of the output value j is estimated with $\frac{N_{i_1 i_2 j}+\epsilon}{N_{i_1 i_2 .}+J\epsilon}$, where $\epsilon = \frac{1}{N}$ is used to avoid zero values. For empty cells, the conditional probability is taken as that of the whole dataset according to $\frac{N_{..j}+\epsilon}{N+J\epsilon}$. A test dataset D_{Test} containing 100,000 instances is generated once for all the experiments, to evaluate the RMSE, according to formula 7.

$$RMSE = \sqrt{\sum_{(x,y)\in D_{Test}} \left(p(y|x) - q(y|x)\right)^2} \tag{7}$$

Figure 9 presents the mean cell number and Figure 10 the average RMSE value of the trained data grid models for each sample size. We also report the RMSE results obtained by a basic parametric model, which assumes that there is exactly one Gaussian vector with independent variables for each output value. This parametric model estimates the means and variances of the Gaussian vectors (overall eight parameters) from the empirical data, which corresponds to a maximum likelihood estimate.

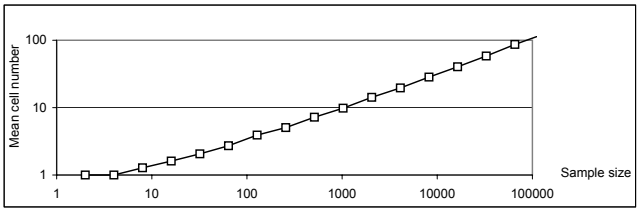


Fig. 9. Mean cell number for the Gaussian mixture pattern dataset, for sample size ranging from 2 to 100,000

The results allow to "quantify" the non asymptotic behavior of the method. Although data grids are non parametric models of conditional density estimation, only 8 instances are sufficient to detect that the data comes more probably from a pattern than from noise. The number of cells steadily grows with the sample size to better approximate the Gaussian mixture pattern. The quality of the approximation, estimated by the RMSE, is of course better with the parametric method, which needs to estimate only 8 parameters instead of more than 100 for data grid models and sample size 100,000.

Let us now focus on the trade-off between the precision (as many cells as possible) and the reliability of the approximation (as many instances per cell as

possible). In the case of this Gaussian pattern, the average number of cells is about $\sqrt{N/8}$ and the average number of instances per cell is about $\sqrt{8N}$, resulting in a $RMSE \approx N^{-1/4}$ (compared to a $RMSE \approx N^{-1/2}$ for the straightforward parametric model). Although these results are only experimental, they provide a quantitative insight on the behavior of the data grid models.

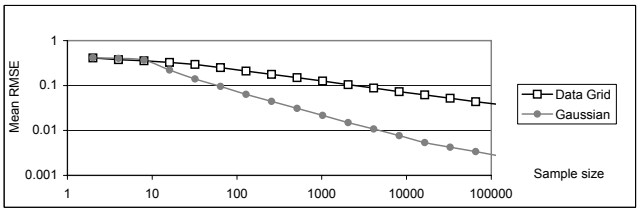


Fig. 10. Mean RMSE value for the Gaussian mixture pattern dataset, for sample size ranging from 2 to 100,000. The results are reported both for the data grid non parametric estimator and the Gaussian mixture parametric estimator

Overall, the method is both resilient to noise and able to detect complex fine grain patterns. It is abel to approximate any conditional data distribution as close as requested, provided that there are enough instances in the train dataset.

5.4 Evaluation of the optimization algorithms

The objective of this section is to evaluate the computational efficiency of the data grid optimization heuristics and to investigate the main components of the algorithms introduced in [Bou07]: greedy bottom-up merge heuristic, metaheuristic and post-optimization.

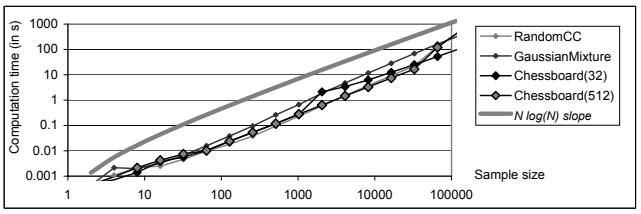


Fig. 11. Computation time for numerical bivariate patterns

The evaluation is performed on a PC with Intel P4 2.5 Ghz processor and 1 Go RAM. Figure 11 reports the average computation time w.r.t the sample size

for four types of numerical patterns extracted from the experiments on artificial datasets. The results confirm that the algorithmic complexity is $O(N \log N)$, as shown by the corresponding slope drawn on the figure.

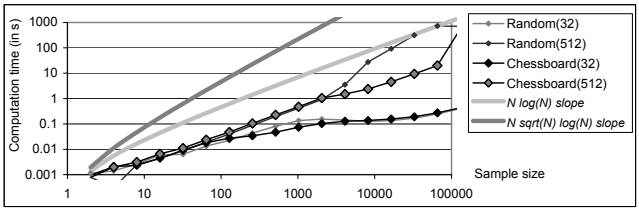


Fig. 12. Computation time for categorical bivariate patterns

Figure 12 reports the same kind of results in the case of categorical patterns. According to the algorithmic study in [Bou07], the computation time should be comprised between $O(N \log N)$ for small numbers of input categorical values and $O(N\sqrt{N} \log N)$ for numbers of values beyond $\sqrt{N}$. This is confirmed by Figure 12, which also shows that the computation time depends both on the number of input values and on the number of instances.

We finally focus on the contribution of each algorithmic component described in [Bou07] to the quality of the optimized data grids. We perform a comparative experiment using the numerical XOR pattern (cf. numerical chessboard pattern of size 2 x 2). The first heuristic evaluated is the greedy bottom-up merge heuristic (GBUM), which is at the heart of the optimization algorithms. The second one is the VNS meta-heuristic which runs the GBUM algorithm several times starting from random data grids of varying size. The third one is the same meta-heuristic, equipped with a post-optimization(VNS+PostOpt) after each run of the GBUM heuristic. This last heuristic is the complete data grid optimization algorithm used in all the experiments of this paper. The evaluation protocol is the same as that of the chessboard pattern experiments. Figure 13 reports the number of cells detected by each algorithm, w.r.t. the sample size.

Surprisingly, the GBUM heuristic needs a very large number of instances to discover the XOR pattern. This can be explained by the sparseness of the initial data grid, which contains N^2 cells, but at most N non empty cells. The GBUM algorithm performs about $2N$ merges between adjacent intervals, but only half of them involve non empty cells. This means that half of the merges are chosen "'randomly"', without being guided by the data. These random merges are likely to destroy the pattern in the early steps of the heuristic, so that the last merges are no longer able to detect the pattern.

The VNS meta-heuristic greatly improves the efficiency of the algorithm, and needs about 100 times less instances than the GBUM heuristic to detect the XOR pattern. The key point is that it starts from initial bivariate discretizations

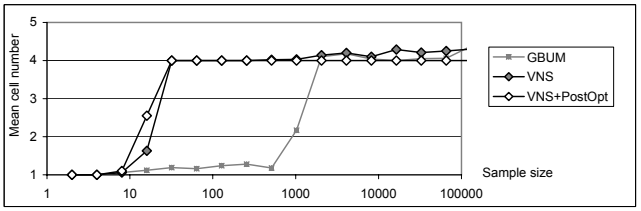


Fig. 13. Comparison of the performance of three data grid optimization algorithms for the detection of an XOR numerical pattern

with fewer intervals, so that the initial data grid contains enough instances per cell to guide the merge process. The drawback is that the boundaries of the patterns are only approximated, since the initial random data grid are likely to miss the correct boundaries. This results in data grids of size 2 x 3, 3 x 2 or 3 x 3 for the XOR pattern. Although the meta-heuristic is partly able to improve this, finding the correct boundaries appears to be too difficult with the VNS randomization strategy. This is illustrated in Figure 13 by an average cell number beyond 4 for the VNS heuristic, especially for large sample sizes. The post-optimization heuristic performs an efficient local search around good initial solutions, and results in the correct detection of the XOR pattern.

The XOR pattern was chosen to be very simple for illustrative reasons. With more complex patterns, the differences of quality between each heuristic are far more important.

6 Evaluation on real datasets

This section evaluates the impact of the MODL bivariate method on supervised classification. The benefits for data preparation are first presented on one example dataset, prior to an intensive evaluation on many datasets.

6.1 Benefits for data preparation

This section evaluates the impact of the bivariate evaluation method on data preparation, especially concerning variable selection and data visualization. The Adult dataset [BM96] is used as an illustrative example. This dataset comes from the US census bureau and contains about 50,000 instances described with 15 input variables (8 categorical and 7 numerical). The objective of the classification task is to predict the output label '> 50K' (rich) or '<= 50K' (poor). The train dataset used in this experiment contains about 70% of the instances.

The univariate analysis is performed using the MODL discretization and value grouping methods. The results are reported in Table 1, where the input variables are sorted by decreasing compression gain g(M). The univariate analysis reveals that two variables, Label and Fnlwgt, are not informative. The two most informative variables are RelationShip and MaritalStatus,

with $g(M) \approx 20\%$. The CapitalGain Variable comes third ($g(M) \approx 13\%$), followed by a group of three variables, Education, Age and EducationNum, with $g(M) \approx 11\%$. This variable ranking provides a first insight on the informative variables, but new questions arise concerning the interactions between these variables. The interaction can be constructive, if there is more information in a pair of variable taken jointly than in the sum of the two univariate informations (like in the XOR pattern). It can be additive if the two variables bring independent information. Redundancy can also be detected if the pair of variables does not bring more information that the most informative variable of the pair. The bi-

Table 1. Univariate ranking of the input variables of the Adult Dataset, sorted by decreasing compression gain g(M)

Rank	g(M)	Variable	Type	Partition size
1	20.8%	Relationship	Categorical	4
2	19.7%	MaritalStatus	Categorical	4
3	13.5%	CapitalGain	Numerical	20
4	11.4%	Education	Categorical	7
5	11.4%	Age	Numerical	7
6	11.2%	EducationNum	Numerical	6
7	9.0%	Occupation	Categorical	7
8	7.1%	HoursPerWeek	Numerical	6
9	5.3%	CapitalLoss	Numerical	15
10	4.6%	Sex	Categorical	2
11	2.2%	Workclass	Categorical	4
12	1.0%	Race	Categorical	2
13	0.7%	NativeCountry	Categorical	3
14	0.0%	Label	Numerical	1
15	0.0%	Fnlwgt	Numerical	1

variate analysis is then performed using the method presented in the paper. The results are reported in Table 2 for the first twenty pairs of variables (among 105 pairs). The two most informative variables RelationShip and MaritalStatus look redundant, since they bring similar information in all the pairs of variables where they are involved. This is confirmed by an inspection of the MaritalStatus x Relationship pair, ranked 20^{th}, whose compression level is only slightly better than that of the Relationship variable taken alone. The redundancy is perfectly detected for the Education x EducationNum pair, since the corresponding data grid, ranked 67^{th}, reduces to a 7 x 1 grid identical to the univariate partition of variable Education. On the opposite, the interaction between Education and MaritalStatus is approximatively additive, since the compression gain of the pair (30.3%) is not far from the sum of the compression gains of each variable

Table 2. Bivariate ranking of the twenty first pairs of input variables of the Adult Dataset, sorted by decreasing compression gain g(M)

Rank	g(M)	Variable 1	Variable 2	Data Grid size
1	31.8%	CapitalGain	Relationship	12 x 4
2	31.1%	CapitalGain	MaritalStatus	15 x 3
3	30.3%	Education	Relationship	7 x 3
4	30.3%	Education	MaritalStatus	7 x 3
5	30.1%	EducationNum	Relationship	6 x 3
6	30.1%	EducationNum	MaritalStatus	7 x 3
7	27.4%	Occupation	Relationship	6 x 4
8	26.7%	MaritalStatus	Occupation	3 x 6
9	24.4%	CapitalLoss	Relationship	9 x 4
10	24.0%	Age	Relationship	5 x 3
11	23.9%	HoursPerWeek	Relationship	5 x 4
12	23.6%	Age	MaritalStatus	5 x 2
13	23.5%	CapitalLoss	MaritalStatus	9 x 3
14	23.5%	HoursPerWeek	MaritalStatus	5 x 3
15	22.6%	Age	CapitalGain	6 x 11
16	21.9%	Relationship	Workclass	4 x 4
17	21.7%	CapitalGain	Education	11 x 6
18	21.6%	CapitalGain	EducationNum	11 x 6
19	21.3%	NativeCountry	Relationship	2 x 4
20	21.1%	MaritalStatus	Relationship	3 x 3

(32.2%). Let us now visualize in Figure 14 the interaction between one numerical variable, EducationNum, and one categorical variable, MaritalStatus. The EducationNum variable is discretized in 7 intervals, with an increasing proportion of rich persons. The MaritalStatus variable is partitioned into three groups of values that look consistent: {Never-Married}, {Divorced, Separated, Widowed} and {Married-civ-spouse, Married-AF-spouse}. The two variables taken jointly bring a meaningful and easily understandable information. The proportion of rich people always increases with the number of education years, but the corresponding curve has not the same shape and is not at the same level according to the marital status. The data grid model can also easily be transformed into an understandable set of rules, with one rule for each cell. For example, the rightmost upper cell in Figure 14 can be described with the following rule: for married people with at least 15 years of education, the proportion of rich is beyond 80%.

Finally, let us focus in Figure 15 on two numerical variables, Age and EducationNum. This pair is ranked 31^{th} with a compression gain of 20.0%. The two input variables are discretized in 7 and 5 intervals (instead of 7 and 6 in the univariate case). The 3D histogram resulting from the data grid model al-

lows a to visualize the interaction between the two input variables in a clearly understandable way.

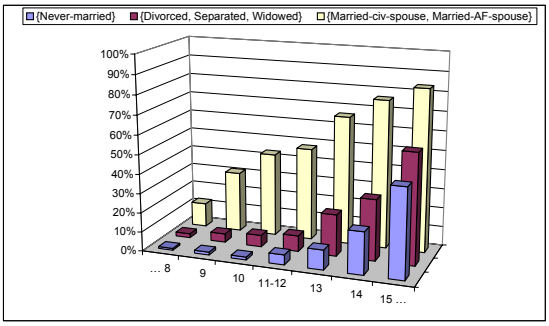


Fig. 14. 3D histogram for the Adult dataset, with the two input variables EducationNum and MaritalStatus on the X and Y axis and the percentage of rich people per cell on the Z axis

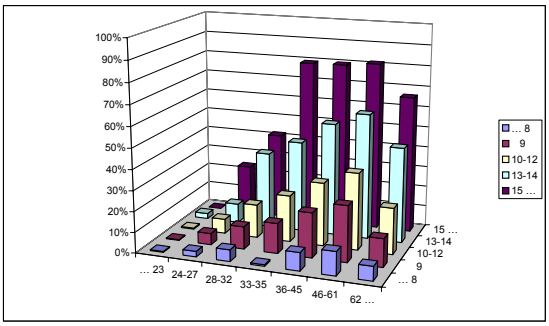


Fig. 15. 3D histogram for the Adult dataset, with the two input variables Age and EducationNum on the X and Y axis and the percentage of rich people per cell on the Z axis

Overall, the bivariate evaluation method is much helpful in the data preparation step of data mining, with ranking of pairs of variables, detection of constructive interactions or of redundancies in the representation space, and easily understandable visualizations of the joint conditional information carried out by each pair of variables.

6.2 Benefits for modeling

In this section, we focus on the predictive accuracy of the data grid models.

In order to evaluate the intrinsic performance of the MODL bivariate method, we introduce a new type of classifier called BestBivariate (B2). This classifier first searches the best pair of variables, which maximizes the probability that its partitioning model explains the target variable. In order to classify a test instance, the input cell related to the instance is retrieved from the learned data grid and the majority target value of this cell is used for prediction. In case where this cell was empty in the learned data grid, the majority class value on the whole train dataset is used for prediction. For sanity check, we also evaluate the BestUnivariate classifier (B1), which proceeds in the same way on the basis of the MODL univariate analysis, and we present the results of the majority classifier (M) which serves as a ground level reference.

In order to evaluate the impact of the method on multivariate classifiers, we evaluate the naive Bayes classifier [LIT92], on the basis of the univariate preprocessing (NB1) and bivariate preprocessing (NB2). We also exploit the enhancements of this classifier described in [Bou06b] [2] , which incorporate both variable selection and model averaging and results in a naive Bayes classifier with weighted variables. The bivariate preprocessing is basically exploited in the experiments, since each bivariate partitioning is simply managed as a constructed variable which expands the data representation space.

The experiments are performed on 30 datasets from the UCI repository [BM96] described in Table 3. They represent a large variety of domains, instance numbers, variable numbers, types of variables (numerical or categorical) and numbers of target values. The test accuracy is evaluated using a stratified ten fold cross-validation. In order to determine whether the performances are significantly different between the SNB2 method and the alternative methods, the t-statistics of the difference of the results is computed, at the 5% confidence level.

6.3 Evaluation results

The results are summarized in Table 4 with the mean of the test accuracy on all the datasets. The number of significant differences for the NB2 classifier is also reported, as well as the average rank of each method. It is noteworthy that the classifier based on one single variable (B1) is as accurate as the best multivariate classifier evaluated in the benchmark (SNB2) in about one quarter of the datasets (no significant differences in 7 datasets out of 30). The classifier that selects only two variables (B2) obtains the best performance in about one third of datasets (10 datasets out of 30).

In order to analyse the results with deeper details, Figure 16 presents the accuracy per dataset for the BestUnivariate, BestBivariate and Naive Bayes

[2] Tool available as a shareware at http://www.francetelecom.com/en/group/rd/offer/software/technologies/middlewares/khiops.html.

Table 3. UCI Datasets

N°	Name	Instances	Numerical variables	Categorical variables	Classes	Majority accuracy
1	Abalone	4177	7	1	28	16.5
2	Adult	48842	7	8	2	76.1
3	Australian	690	6	8	2	55.5
4	Breast	699	10	0	2	65.5
5	Crx	690	6	9	2	55.5
6	German	1000	24	0	2	70.0
7	Glass	214	9	0	6	35.5
8	Heart	270	10	3	2	55.6
9	Hepatitis	155	6	13	2	79.4
10	HorseColic	368	7	20	2	63.0
11	Hypothyroid	3163	7	18	2	95.2
12	Ionosphere	351	34	0	2	64.1
13	Iris	150	4	0	3	33.3
14	LED	1000	7	0	10	11.4
15	LED17	10000	24	0	10	10.7
16	Letter	20000	16	0	26	04.1
17	Mushroom	8416	0	22	2	53.3
18	PenDigits	7494	16	0	10	10.4
19	Pima	768	8	0	2	65.1
20	Satimage	6435	36	0	6	23.8
21	Segmentation	2310	19	0	7	14.3
22	SickEuthyroid	3163	7	18	2	90.7
23	Sonar	208	60	0	2	53.4
24	Spam	4307	57	0	2	64.7
25	Thyroid	7200	21	0	3	92.6
26	TicTacToe	958	0	9	2	65.3
27	Vehicle	846	18	0	4	25.8
28	Waveform	5000	21	0	3	33.9
29	Wine	178	13	0	3	39.9
30	Yeast	1484	8	1	10	31.2

Table 4. Mean of the test accuracy, number of significant differences for the NB2 classifier and average rank of each classifier on 30 UCI datasets

	SNB2	NB2	SNB1	NB1	B2	B1	M
Mean	83.9%	81.9%	82.4%	81.4%	73.4%	67.6%	48.5%
Win/Draw/Loss		15/15/0	12/18/0	14/16/0	20/10/0	23/7/0	
Average rank	1.8	3.3	2.3	3.4	4.4	5.4	

classifiers, relatively to the accuracy of the majority classifier. The BestBivariate classifier is always more accurate than the BestUnivariate classifier, which confirms the capacity of the bivariate evaluation method to efficiently select a predictive pair of variables. However, the BestBivariate classifier is significantly dominated by the Naive Bayes classifier, which exploits the whole set of variables.

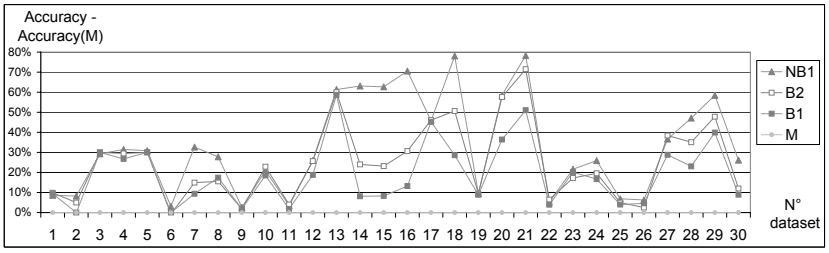


Fig. 16. Difference of accuracy between each evaluated classifier and the majority classifier (M). The evaluated classifiers are the BestUnivariate (B1), BestBivariate (B2) and Naive Bayes (NB1) classifiers

Figure 17 focuses on the Naive Bayes multivariate classifier, and studies the impact of exploiting or not the pairs of variables (NB2 and NB1) and of that of variable selection (SNB2 and SNB1). Using the pairs of variables enlarges the representation space, which potentially allows to detect new predictive information. On the other hand, redundancies in the univariate representation are multiplied in the bivariate representation, which is detrimental to the Naive Bayes assumption. Figure 17 shows that the two effects are observed on the datasets of the experiments, with significant loss of accuracy for datasets 1, 2, 6, 9, 22, 26, and strong gain of accuracy for datasets 16, 18, 20, 23, 27. The variable selection method [Bou06b] used in the SNB1 classifier confirms its beneficial impact on test accuracy, systematic but slight, compared to the NB1 classifier. When efficient variable selection is used together with the pairs of variables (SNB2), the gain in accuracy becomes both important, with an average improvement of

2.5% (15% for the Letter dataset), and highly significative, with 14 significant wins and 0 loss.

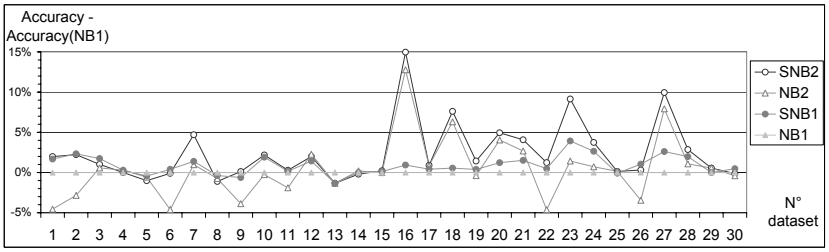


Fig. 17. Difference of accuracy between each evaluated classifier and the Naive Bayes classifier (NB1). The evaluated classifiers are the Naive Bayes classifier exploiting all pairs of variables (NB2) and the Selective Naive Bayes classifiers based on univariate preprocessing (SNB1) or bivariate preprocessing (SNB2)

7 Conclusion

The bivariate evaluation method introduced in this paper is based on a partitioning model of each input variables, in intervals for numerical variables and in groups of values for categorical variables. The cross-product of the univariate partitions, called a data grid, allows to quantify the conditional information relative to the output variable. The data grid models are evaluated using a MAP approach, and the best joint partitioning is searched in the model space owing to efficient heuristics.

Our method is non parametric both in the statistical and algorithmic sense : it does not rely on any statistical hypothesis for the data distribution (like Gaussianity for instance) and, as the criterion is regularized, there is no parameter to tune before optimizing it. This strong point enables to consider large datasets.

The data grid models are non asymptotic universal approximators of the class conditional density for pairs of input variables. Experiments on artificial datasets show that the method is both very resilient to noise and able to detect complex fine grain patterns, even with few instances. This requires sophisticated algorithms, such as those described in [Bou07]. The experiments confirm that these algorithms run in $O(N \log N)$ time and that less complex algorithms fail to be as efficient.

The benefit of data grid models for data exploration is evaluated as a case study on a real dataset. The results demonstrate the ability of the method to detect constructive interactions or, on the opposite, redundancies between the input variables, and highlight the vizualisation and data understanding capacities of the data grids.

The impact of bivariate preprocessing on classification accuracy is evaluated through extensive experiments on 30 UCI datasets. The results show that the bivariate evaluation method is able to select strongly predictive pairs of variables. However, the average impact on classification accuracy is not conclusive for the Naive Bayes classifier when all the pairs of variables are exploited. The problem is that the potential benefit of additional predictive information in the pairs of variables is balanced by the detrimental effect of increase redundancies in the presentation space. When the Naive Bayes classifier is equipped with an efficient variable selection method, the benefit of bivariate preprocessing becomes both systematic and important: the classification accuracy always increases, with significant differences in half of the cases.

References

[Bay01] S. Bay. Multivariate discretization for set mining. *Machine Learning*, 3(4):491–512, 2001.

[BFOS84] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. California: Wadsworth International, 1984.

[BM96] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1996. http://www.ics.uci.edu/mlearn/MLRepository.html.

[Bou05] M. Boullé. A Bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research*, 6:1431–1452, 2005.

[Bou06a] M. Boullé. MODL: a Bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1):131–165, 2006.

[Bou06b] M. Boullé. Regularization and averaging of the selective naïve Bayes classifier. In *International Joint Conference on Neural Networks*, pages 2989–2997, 2006.

[Bou07] M. Boullé. Optimization algorithms for bivariate evaluation of data grid models. *Advances in Data Analysis and Classification*, 2007. submitted.

[CCK+00] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. *CRISP-DM 1.0 : step-by-step data mining guide*, 2000.

[CLNL87] D.B. Carr, R.J. Littlefield, W.L. Nicholson, and J.S. Littlefield. Scatterplot matrix techniques for large n. *Journal of the American Statistical Association*, 82:424–436, 1987.

[FGG97] N. Friedman, D. Geiger, and M. Goldsmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[FI92] U. Fayyad and K. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8:87–102, 1992.

[GE03] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[GGHD06] I. Guyon, S. Gunn, A. Ben Hur, and G. Dror. *Feature Extraction: Foundations And Applications*, chapter 9, pages 237–263. Springer, 2006. Design and Analysis of the NIPS2003 Challenge.

[Han80] T.S. Han. Multiple mutual informations an multiple interactions in frequency data. *Information and Control*, 46(1):26–45, July 1980.

[HM01] P. Hansen and N. Mladenovic. Variable neighborhood search: principles and applications. *European Journal of Operational Research*, 130:449–467, 2001.

[Kas80] G.V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29(2):119–127, 1980.

[KJ97] R. Kohavi and G. John. Wrappers for feature selection. *Artificial Intelligence*, 97(1-2):273–324, Décembre 1997.

[KK99] W. Kwedlo and M. Kretowski. An evolutionary algorithm using multivariate discretization for decision rule induction. In *Principles of Data Mining and Knowledge Discovery*, pages 392–397, 1999.

[Kon91] I. Kononenko. Semi-naive Bayesian classifier. In Y. Kodrato, editor, *Sixth European Working Session on Learning (EWSL91)*, volume 482 of *LNAI*, pages 206–219. Springer, 1991.

[LIT92] P. Langley, W. Iba, and K. Thompson. An analysis of Bayesian classifiers. In *10th national conference on Artificial Intelligence*, pages 223–228. AAAI Press, 1992.

[MC99] S. Monti and G.F. Cooper. A latent variable model for multivariate discretization. In *The Seventh International Workshop on Artificial Intelligence and Statistics*, pages 249–254, 1999.

[McG54] W.J. McGill. Multivariate information transmission. *IEEE Trans. Information Theory*, 4(4):93–111, 1954.

[NG05] M. Nadif and G. Govaert. Block clustering of contingency table and mixture model. volume 3646 of *LNCS*, pages 249–259, 2005.

[Pyl99] D. Pyle. *Data preparation for data mining*. Morgan Kaufmann Publishers, Inc. San Francisco, USA, 1999.

[Qui93] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[Sap90] G. Saporta. *Probabilités analyse des données et statistique*. Technip, 1990.

[Sha48] C.E. Shannon. A mathematical theory of communication. Technical report, Bell systems technical journal, 1948.

[SJ04] H. Steck and T. Jaakkola. Predictive discretization during model selection. *Pattern Recognition*, LNCS 3175:1–8, 2004.

[VR03] R. Vilalta and I. Rish. A decomposition of classes via clustering to explain and improve naive Bayes. In *Proceedings of the 14th European Conference on Machine Learning*, pages 444–455, 2003.

[WBW05] G.I. Webb, J.R. Boughton, and Z. Wang. Not so naive bayes: Aggregating one-dependence estimators. *Machine Learning*, 58(1):5–24, 2005.

[ZR00] D.A. Zighed and R. Rakotomalala. *Graphes d'induction*. Hermes, France, 2000.

[ZRES05] D.A. Zighed, G. Ritschard, W. Erray, and V.M. Scuturici. Decision trees with optimal joint partitioning. *International Journal of Intelligent System*, 20(7):693–718, 2005.

D

Optimization Algorithms for Bivariate Evaluation of Data Grid Models

Optimization Algorithms for Bivariate Evaluation of Data Grid Models

Marc Boullé

France Télécom R&D Lannion,
marc.boulle@orange-ftgroup.com

Abstract. In the domain of data preparation for supervised learning, many ranking methods have been proposed to assess the predictive information of individual input variable. For example, supervised discretization methods partition the input variable into a set of intervals, which allows to evaluate the correlation between the input and output variables. Such partitions can be extended to the bivariate case, owing to a partition of two input variables, in intervals in the numerical case and in groups of values in the categorical case. The resulting input data grid allows a joint evaluation of the two input variables with respect to the output variable. In this paper, we introduce a family of additive evaluation criteria for data grid models and present new algorithms which efficiently search the model space.

1 Introduction

In [Bou07], we have introduced bivariate partitioning models called data grids in order to quantify the predictive importance of pairs of input variables in supervised learning. These models can be evaluated owing to an analytic criterion, which is Bayes optimal but is difficult to optimize. The purpose of this paper is to describe efficient algorithms for the optimization of data grid models. Their evaluation on artificial and real data is reported in [Bou07].

Many discretization methods have been proposed in the literature to evaluate the correlation between a numerical input variable and an output variable [Cat91,Hol93,DKS95,ZR00,LHTD02]. Some discretization methods such as ChiMerge [Ker91] or MDLPC [FI92] evaluate a bipartition of one interval into two sub-intervals and apply the method recursively. Since the criterion evaluates only a local decision for two adjacent intervals, the discretization of the whole numerical domain cannot be optimized globally. Other discretization methods such as BalancedGain [KBR84], Fusinter [ZRR98] or MODL [Bou06] exploit a global criterion which evaluates the complete discretization, so that looking for an optimal discretization makes sense.

For some classes of global criteria such as additive criteria, an optimal algorithm based on dynamic programming [TKS95,ER96] allows to find the optimal discretization in $O(N^3)$, where N is the number of instances. A more practical

time complexity is achievable using top-down or bottom-up heuristics. Top-down methods start from the complete numerical domain interval and recursively split it into smaller intervals. Bottom-up methods start from the set of single value intervals and iteratively merge neighboring intervals. In the case of global criteria, each evaluation of a discretization requires $O(N)$ time since the discretization contains $O(N)$ intervals. At each step of the heuristic, there are $O(N)$ splits or merges to evaluate, and the number of steps is $O(N)$. Overall, a straightforward implementation of the heuristic runs in $O(N^3)$ time. However, in case of additive criteria, computation time can be saved provided that intermediate evaluation results are kept into memory. Using a careful implementation such as that described in [Bou04], the algorithm runs in $O(N \log N)$ time.

While discretization methods look for a partition of a numerical variable into intervals, value grouping methods search a partition of categorical variable into groups of values. Value grouping methods exploit either local criteria such as CHAID [Kas80] or global criteria such as Gain Ratio [Qui93]. Even with additive evaluation criteria, no optimal algorithm is available in the literature, since any partition of the input values is possible for value grouping. The time efficient bottom-up discretization heuristic can be applied to value grouping, but its time complexity is $O(N^2 \log N)$ since $O(N^2)$ merge decisions have to be evaluated for very large numbers of input values. In [Bou05], the bottom-up standard heuristic is enhanced with pre-processing and post-processing steps, in order to reach a practical $O(N \log N)$ time complexity without sacrifying the quality of the solution.

The univariate partitioning of an input variable has been extended to the bivariate case in [Bou07], using data grid models. Each input variable is partitioned, in intervals in the numerical case and in groups of values in the categorical case. The resulting input data grid allows a joint evaluation of the two input variables with respect to the output variable. The optimization of a data grid is a combinatorial problem. Let us first focus on numerical input variables. We describe in Algorithm 1 an adaptation of the greedy bottom up merge heuristic (GBUM) to optimize the data grids. The method starts with the maximum data grid M_{Max}, which corresponds to the finest possible univariate discretizations, based on single value intervals. It evaluates all the merges between adjacent intervals, and perform the best merge if the evaluation criterion decreases after the merge. The process is reiterated until no further merge can decrease the criterion.

Each evaluation of a data grid requires $O(N^2)$ time, since the initial data grid model M_{Max} contains N^2 cells. Each step of the algorithm relies on $O(N)$ evaluations of interval merges, and there are at most $O(N)$ steps, since the data grid reduces to the null model $M_\emptyset$ once all the possible merges have been performed. Overall, the time complexity of the algorithm is $O(N^4)$ using a straightforward implementation of the algorithm.

We introduce in section 2 a family of additive criteria for data grid models and describe in section 3 an algorithmic structure that exploits the sparseness of the data in the bidimensional space. We show in section 4 how to exploit this

Algorithm 1 Greedy Bottom Up Merge heuristic (GBUM)

Require: M {Initial data grid solution}
Ensure: $M^*, c(M^*) \leq c(M)$ {Final solution with improved cost}
1: $M^* \leftarrow M$
2: **while** improved solution **do**
3: **for all** Merge m between two parts of variable X_1 or X_2 **do**
4: $M' \leftarrow M^* + m$ {Evaluate merge m on data grid M^*}
5: **if** $c(M') < c(M^*)$ **then**
6: $M^* \leftarrow M'$
7: **end if**
8: **end for**
9: **end while**

algorithmic structure to implement the GBUM heuristic in $O(N \log N)$ time for additive criteria in the case of numerical input variables. We present in section 5 several post-optimization heuristics, that still run in $O(N \log N)$ time. While post-optimizations may help to refine a good solution, the main heuristic may be unable to obtain such an initial good solution. We propose to tackle this problem using a meta-heuristic described in section 6, which mainly benefits from multiple runs of the algorithms with different random initial solutions. We extend the algorithm to handle categorical input variables in section 7. Finally, the algorithms are summarized in section 8.

2 Additive data grid evaluation criterion

This section formally states the definitions and notations related to data grid models and introduces a family of additive evaluation criteria, first in the case of univariate models, then in the case of bivariate data grid models.

2.1 Data grid models

Let us first focus on the univariate case and formalize in Definition 1 the univariate partitioning models introduced for the discretization of numerical variables [Bou06] and for the grouping of values of categorical variables [Bou05].

Definition 1. *An univariate partitioning model is defined by a set of intervals (resp. groups of values) and by the distribution of the output values in each interval (resp. group of values).*

Notations.

- Y: output variable,
- X: input variables,
- N: number of instances,
- J: number of output values,

- V: number of values(in the categorical case),
- I: number of parts (intervals or groups of values),
- $N_{i.}$: number of instances in the input part i,
- N_{ij}: number of instances of output value j in the input part i.

We now present in Definition 2 the data grid models introduced in [Bou07]. They consist in a family of bivariate partitioning models, the purpose of which is to evaluate the joint information between a pair of input variables and an output categorical variable. The input variables can be of any type, numerical or categorical.

Definition 2. *A data grid model is a bivariate partitioning model defined by a partition of each input variable, in intervals in the numerical case or groups of values in the categorical case, and by the distribution of the output values in each cell of the data grid resulting from the cross-product of the univariate partitions.*

The components of a *data grid* model are the input *variables*, the *parts* (intervals or groups of values) in the univariate partitions, and the *cells* (cross-product of two parts). An illustrative instance of data grid model is given in Figure 1.

Notations.

- Y: output variable,
- X_1, X_2: input variables,
- N: number of instances,
- J: number of output values,
- V_1, V_2: number of values for each input variable (in the categorical case),
- I_1, I_2: number of parts for each input variable,
- $N_{i_1..}$: number of instances in the part i_1 of variable X_1,
- $N_{.i_2.}$: number of instances in the part i_2 of variable X_2,
- $N_{i_1 i_2.}$: number of instances in the input data cell (i_1, i_2),
- $N_{i_1 i_2 j}$: number of instances of output value j in the input data cell (i_1, i_2).

A data grid model describes the distribution of the output values given the input values. It is completely defined by the numbers of parts I_1 and I_2, the specification of the univariate partitions which results in part frequencies $\{N_{i_1..}\}$ and $\{.N_{i_2.}\}$ and the distribution of the output values $\{N_{i_1 i_2 j}\}$ in each cell (i_1, i_2) of the data grid . It is noteworthy that the cell frequencies $\{N_{i_1 i_2.}\}$ do not belong to the parameters of the data grid models: they are derived from the specification of the two univariate partitions and from the dataset.

2.2 Univariate case

In the univariate case (discretization or value grouping), the considered models consist in a partition of one input variable X in I parts. We introduce in definition 3 a family of additive criteria to evaluate such partitions.

Definition 3. *An evaluation criterion $c(M)$ of a univariate partition model M is additive if it can be decomposed as a sum of the following terms:*
- *a variable criterion $c^{(V)}(X, I)$, which relies only on features of the input variable X and on the number of parts I in the partition,*
- *a part criterion $c^{(P)}(P_i)$ for each part P_i of the univariate partition of the input variable X, which relies only on features of the part.*

An additive univariate partition evaluation criterion can be written like in formula 1. In the rest of the paper, $C(M)$ is referred to as the cost of M.

$$c(M) = c^{(V)}(X, I) + \sum_{i=1}^{I} c^{(P)}(P_i) \tag{1}$$

For example, the discretization criterion [Bou06] is additive with

$$c^{(V)}(X, I) = \log N + \log \binom{N + I - 1}{I - 1},$$
$$c^{(P)}(P_i) = \log \binom{N_{i\cdot} + J - 1}{J - 1} + \log \frac{N_{i\cdot}!}{N_{i1}! N_{i2}! \dots N_{iJ}!}.$$

The value grouping criterion [Bou05] is also additive with

$$c^{(V)}(X, I) = \log V + \log B(V, I),$$
$$c^{(P)}(P_i) = \log \binom{N_{i\cdot} + J - 1}{J - 1} + \log \frac{N_{i\cdot}!}{N_{i1}! N_{i2}! \dots N_{iJ}!}.$$

We now describe in property 1 the impact of a merge between two parts for additive evaluation criteria.

Property 1. Let M be a univariate partition model of variable X, P_{i_a} and P_{i_b} two parts of the partition and M' the partition resulting from the merge of the two parts. Then the variation of the evaluation criterion $\delta c(M', M) = c(M') - c(M)$ can be decomposed as a sum of one variation term for the variable criterion and one variation term for the part criterion.

More formally, we have

$$\delta c(M', M) = \delta c^{(V)}(X, I) + \delta c^{(P)}(P_{i_a}, P_{i_b}), \tag{2}$$

with $\delta c^{(V)}(X, I) = c^{(V)}(X, I - 1) - c^{(V)}(X, I)$ and
$\delta c^{(P)}(P_{i_a}, P_{i_b}) = c^{(P)}(P_{i_a} \cup P_{i_b}) - c^{(P)}(P_{i_a}) - c^{(P)}(P_{i_b})$.

Property 1 means that each merge has only a local impact on the criterion. In the case of discretization, all the merges can thus be evaluated in $\mathrm{O}(N)$ time since each interval is involved in at most two merges, each of which evaluated in $\mathrm{O}(1)$. This key property is at the ground of the optimized version of the univariate greedy heuristic [Bou06], which runs in $\mathrm{O}(N \log N)$ time (and $\mathrm{O}(N)$ memory complexity) instead of $\mathrm{O}(N^3)$ time with a straightforward implementation.

2.3 Bivariate case

We now extend in definition 4 the notion of additive criterion to data grid models, which rely on a univariate partition of the values for each input variable, and for each cell of the cross-product of the univariate partitions, on the distribution of the output values.

Definition 4. *An evaluation criterion $c(M)$ of a data grid model M is additive if it can be decomposed as a sum of the following terms:*

- *a grid criterion $c^{(G)}(G)$, which relies only on the number $G = I_1 I_2$ of cells in the data grid,*
- *a variable criterion $c^{(V)}(X_k, I_k)$, which relies only on features of the input variable X_k ($k \in \{1,2\}$) and on the number of parts I_k of its partition,*
- *a part criterion $c^{(P)}\left(P_{i_k}^{(k)}\right)$ for each part $P_{i_k}^{(k)}$ of the univariate partition of the input variable X_k ($k \in \{1,2\}$), which relies only on features of the part,*
- *a cell criterion $c^{(V)}(C_{i_1 i_2})$ for each cell $C_{i_1 i_2}$ of the data grid, which relies only on features of the cell, and which is null for empty cells.*

An additive data grid evaluation criterion can be written like in formula 3.

$$\begin{aligned} c(M) &= c^{(G)}(G) + \sum_{k\in\{1,2\}} c^{(V)}(X_k, I_k) \\ &+ \sum_{k\in\{1,2\}} \sum_{i_k=1}^{I_k} c^{(P)}\left(P_{i_k}^{(k)}\right) + \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2} c^{(C)}(C_{i_1 i_2}) \end{aligned} \tag{3}$$

For example, in the case of two numerical input variables, the evaluation criterion [Bou07] is additive with

$$\begin{aligned} c^{(G)}(G) &= 0, \\ c^{(V)}(X_1, I_1) &= \log N + \log \binom{N + I_1 - 1}{I_1 - 1}, \\ c^{(V)}(X_2, I_2) &= \log N + \log \binom{N + I_2 - 1}{I_2 - 1}, \\ c^{(P)}\left(P_{i_1}^{(1)}\right) &= 0, \\ c^{(P)}\left(P_{i_2}^{(2)}\right) &= 0, \\ c^{(C)}(C_{i_1 i_2}) &= \log \binom{N_{i_1 i_2} + J - 1}{J - 1} + \log \frac{N_{i_1 i_2 .}!}{N_{i_1 i_2 1}! N_{i_1 i_2 2}! \ldots N_{i_1 i_2 J}!}. \end{aligned}$$

Since the number of instances N and of output values J are supposed to be known, they may be used in every component of the additive criterion. On the opposite, the features of variables, parts and cells are local to their component: they mainly consist in the frequency and frequency per output value locally to

the component (for example, $N_{i_1 i_2 .}$ and $N_{i_1 i_2 j}$ frequencies in the cells). The other data grid evaluation criteria [Bou07] in the case of two categorical variables or mixed type variables are also additive.

We now describe in property 2 the impact of a merge between two parts of the same variable in a data grid model for additive evaluation criteria. For sake of simplicity, we consider the variable X_1, without loss of generality.

Property 2. Let M be a data grid model, $P^{(1)}_{i_{1_a}}$ and $P^{(1)}_{i_{1_b}}$ two parts of the univariate partition of variable X_1 and M' the data grid resulting from the merge of the two parts. Then the variation of the evaluation criterion $\delta c(M', M) = c(M') - c(M)$ can be decomposed as a sum of the variation of the criterion for each component of the data grid involved in the merge.

More formally, we have

$$\begin{aligned}\delta c\left(M', M\right) = {} & \delta c^{(G)}\left(G, I_1\right) + \delta c^{(V)}\left(X_1, I_1\right) \\ & + \delta c^{(P)}\left(P_{i_a}, P_{i_b}\right) + \sum_{i_2=1}^{I_2} \delta c^{(C)}\left(C_{i_{1_a} i_2}, C_{i_{1_b} i_2}\right),\end{aligned} \tag{4}$$

$$\begin{aligned}\text{with } \delta c^{(G)}\left(G, I_1\right) &= c^{(G)}\left(G\frac{I_1 - 1}{I_1}\right) - c^{(G)}\left(G\right), \\ \delta c^{(V)}\left(X_1, I_1\right) &= c^{(V)}\left(X_1, I_1 - 1\right) - c^{(V)}\left(X_1, I_1\right), \\ \delta c^{(P)}\left(P^{(1)}_{i_{1_a}}, P^{(1)}_{i_{1_b}}\right) &= c^{(P)}\left(P^{(1)}_{i_{1_a}} \cup P^{(1)}_{i_{1_b}}\right) - c^{(P)}\left(P^{(1)}_{i_{1_a}}\right) - c^{(P)}\left(P^{(1)}_{i_{1_b}}\right) \text{ and} \\ \delta c^{(C)}\left(C_{i_{1_a} i_2}, C_{i_{1_b} i_2}\right) &= c^{(C)}\left(C_{i_{1_a} i_2} \cup C_{i_{1_b} i_2}\right) - c^{(C)}\left(C_{i_{1_a} i_2}\right) - c^{(C)}\left(C_{i_{1_b} i_2}\right).\end{aligned}$$

Given Property 2, the evaluation of a merge can be performed in O(1) for the data grid, variable and part components of the criterion and in O(1) per non empty cell involved in the merge.

3 Algorithmic structure for data grid optimization

In this section, we present an algorithmic structure designed to efficiently optimize data grids. This structure exploits the sparseness of the data in the bivariate case. Figure 1 shows a data grid for the (V1, V7) variables of the Wine dataset [BM96], with about half of the cells being empty. In the extreme case, the maximum data grid M_{Max} have O(N) parts in each univariate discretization and O(N^2) cells. However, this maximum data grid contains at most N non-empty cells, since the number of such cells is below the number of instances in the dataset.

We define below and illustrate in Figure 2 the components of an algorithmic structure that allows an efficient storage of data grids.

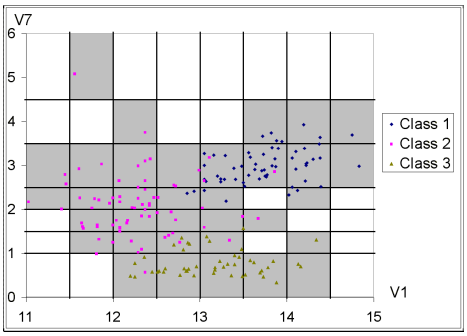


Fig. 1. Data grid for the input variables V1 and V7 of the Wine dataset. The V1 variable is discretized into 8 parts and the V7 variable into 7 parts. There are 34 non empty cells among the 56 cells of the data grid

- **Data Grid**: main data grid component, which collects the data grid features and contains a list a *Variable* components and a set of *Cell* components,
- **Variable**: sub-component of a *Data Grid*, which collects the variable features and contains a list of *Part* components,
- **Part**: sub-component of a *Variable*, which collects the part features and references a list of *Cell* components,
  - **Interval**: part in the case of a numerical variable,
  - **Value Group**: part in the case of a categorical variable,
- **Cell**: cell defined by its *Signature* (a pair of *Parts*).

In the rest of the paper, we will refer to the data grid concepts (variable, part, cell) using lower case characters and to the algorithmic components (Data Grid, Variable, Part, Cell) with a leading uppercase character.

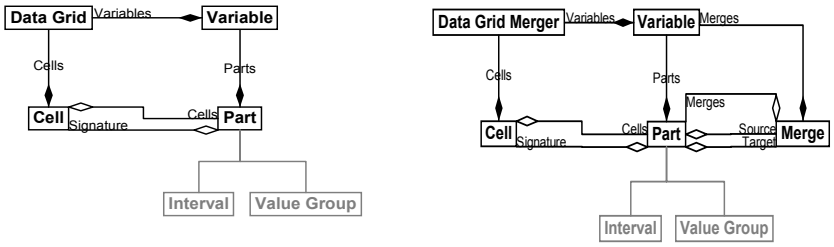


Fig. 2. Algorithmic structure Data Grid defined to store a data grid and its extension Data Grid Merger defined to store all the part merges considered by the bottom-up merge heuristic

For example, the data grid shown in Figure 1 contains 2 variables and 34 non-empty cells. The first variable contains 8 parts and the second one 7 parts. The first part of the V1 variable (interval $]-\infty, 11.5]$) references two cells, whose signatures are $(]-\infty, 11.5],]2, 2.5])$ and $(]-\infty, 11.5],]2.5, 3])$.

We now extend the Data Grid structure to the **Data Grid Merger** structure, which contains the list of all the possible part **Merges**. As shown in Figure 2, each variable contains the list of all the merges between two parts. Each merge references its *Source* part and its *Target* part, and each part maintains the list of all its related merges.

Finally, let us introduce two last terms which are useful to describe the optimization algorithms. Each merge corresponds to one input variable, called the *inner* variable. The other input variable is called the *outer* variable. Similarly, we relate to inner or outer parts (or to inner or outer merges), according to whether these components belong to the inner or outer variable.

4 Optimized implementation of the greedy bottom-up heuristic

In this section, we focus on numerical input variables. The case of categorical variables is discussed in section 7.

The purpose of this section is to demonstrate that the Greedy Bottom Up Merge (GBUM) heuristic (Algorithm 1) can run in $\mathrm{O}(N \log N)$ time, owing to the additivity of the criterion introduced in section 2 and to the Data Grid Merger algorithmic structure defined in section 3. We detail the main subroutines necessary to achieve this time complexity: initialization of the Data Grid structure in section 4.1, evaluation of all the possible merges and initialization of the Data Grid Merger structure in section 4.2, maintenance of this structure throughout the GBUM algorithm in sections 4.3 and 4.4. We summarize the algorithmic complexity results in section 4.5.

4.1 Initialization of the Data Grid structure

The Data Grid initialization subroutine is presented in Algorithm 2. Each instance in the dataset is stored in the Data Grid with at most one Cell and two Parts. Thus, the memory requirement of a Data Grid is $\mathrm{O}(N)$, like that of the dataset.

During the initialization, the Parts are first created, ranked by their input values. This requires a sort of the dataset for each variable, in $\mathrm{O}(N \log N)$ time. The cells can be efficiently initialized owing to a lookup table for the Parts of each Variable (based on a hash function of the input values) and a lookup table for the Cells of the Data Grid (based on a hash function of the cell signatures). These hash tables allow to create or retrieve each Cell in $O(1)$. Overall, the time complexity of the Data Grid initialization subroutine is $\mathrm{O}(N \log N)$.

Algorithm 2 Subroutine: initialization of a Data Grid Structure

Require: *Dataset* {Data in its tabular format (instances*variables)}
Ensure: *Data Grid* {Data in its Data Grid format}
Create an empty *Data Grid*
for all input variable $V \in \{X_1, X_2\}$ **do**
  Create an empty *Variable* V in the *Data Grid*
  Sort *Dataset* according to V
  Create initial *Parts* (*Interval* or *Value Group*) for each value of V
end for
for all instance in *Dataset* **do**
  Lookup the corresponding *Part* for each input *Variable*
  Compute the *Signature* of the corresponding *Cell*
  Lookup the *Cell* corresponding to this *Signature*
  if the *Cell* does not exist **then**
    Create the *Cell* in the *Data Grid*
  end if
  Update the *Cell* output value frequencies
end for

4.2 Initialization of the Data Grid Merger structure

The Data Grid Merger initialization subroutine is described in Algorithm 3. Since only adjacent intervals can be merged, there are at most $O(N)$ possible merges for each numerical input variable. Thus, the memory requirement of a Data Grid Merger is $O(N)$.

The main loop in the algorithm evaluates the local cost variation resulting from each merge, and takes benefit from the additivity of the evaluation criterion. The cost variation can be evaluated once at the variable level ($\delta c^{(V)}$) and at the data grid level ($\delta c^{(G)}$), since each merge results in the same new part number (see Property 2). On the opposite, the cost variation has to be evaluated locally to each merge at the part level ($\delta c^{(P)}$) for the two parts involved in the merge and at the cell level ($\delta c^{(C)}$) for the non-empty cells involved in merge. We show in Algorithm 4 that the merge evaluation subroutine requires a computation time linear in the number of cells involved in the merge. Since each cell participates to at most two merges (with the preceding and following interval), the overall computation time for all the merges is $O(N)$. Last, the merges are sorted according to their cost variation, in order to retrieve the most interesting merge. This sort requires $O(N \log N)$ time. Overall, the time complexity of the Data Grid Merger initialization subroutine is $O(N \log N)$.

We now comment the merge evaluation subroutine described in Algorithm 4. When the source and target parts are merged, the routine mainly evaluates the impact of the merge on the cells. Three situations may occur, as illustrated in Figure 3. In the first situation, the source cell collides with a target cell. This situation, identified in $O(1)$ per cell owing to the hash function of the cell signatures, has an impact on the cost variation related to the merge. In the two other situations, there is either no target cell corresponding to a source cell,

Algorithm 3 Subroutine: initialization of a Data Grid Merger Structure

Require: *Dataset* {Data in its tabular format (instances*variables)}
Ensure: *Data Grid Merger* {Data in its Data Grid Merger format}
Call subroutine 2 {Initialize the core Data Grid structure}
Initialize the cost of each component (*Data Grid, Variable, Part, Cell*)
for all *Variable* in the *Data Grid* **do**
  Compute the local variation of cost ($\delta c^{(V)}$) resulting from one part less in the variable partition
  for all *Part* **do**
    for all Possible merge between the *Part* and an adjacent *Part* **do**
      Create an empty *Merge*
      Link the *Merge* to each of its *Parts*
      Call subroutine 4 {Evaluation of the merge}
    end for
  end for
  Sort the *Merges* by decreasing cost variation and store them the *Variable* using a sortable list
end for

either no source cell corresponding to a target cell. This has no impact on the cost variation, since the involved cells are the same before and after the merge.

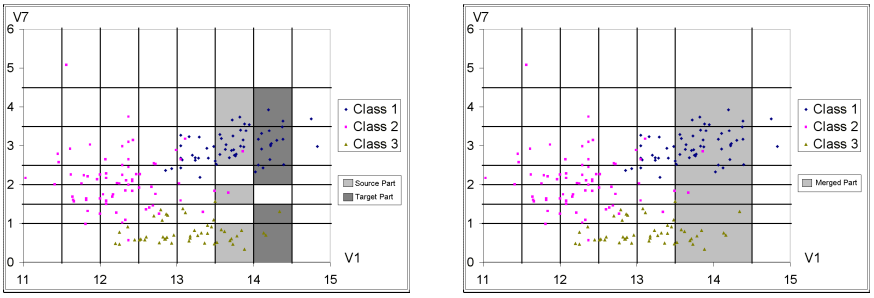


Fig. 3. Diagrams representing the merge between the source part $[13.5, 14.0[$ and the target part $[14.0, 14.5[$ of variable V_1 in the Wine dataset, before the merge (on the left) and after the merge (on the right). Three sources cells collide with three target cells, one source cell has no related target cell and two target cells have no related source cells

4.3 Completion of the merges: impacts on the Data Grid

Once the Data Grid Merger structure is initialized, the best merge per input variable can be retrieved in O(1) from the related sorted list of merges. Using the additivity of the evaluation criterion, we just have to add the cost variation related to the variable ($\delta c^{(V)}$) and to the data grid ($\delta c^{(G)}$) to retrieve the

Algorithm 4 Subroutine: evaluation of the local cost variation resulting from a Merge

Require: *Merge* $M^{(1)}$
{We assume that the inner variable of the merge is variable X_1 and that the number of cells is less in its source part than in its target part}
Ensure: Evaluation of $M^{(1)}$ $\{\delta c = \delta c^{(P)} + \sum \delta c^{(C)}\}$
$P_s^{(1)} \leftarrow$ source part of $M^{(1)}$
$P_t^{(1)} \leftarrow$ target part of $M^{(1)}$
$\delta c \leftarrow \delta c^{(P)}(P_s^{(1)}, P_t^{(1)})$
for all *Cell* $C \in P_s^{(1)}$ **do**
  $P^{(2)} \leftarrow$ outer part of C
  $s \leftarrow (P_t^{(1)}, P^{(2)})$ {compute the signature s of C' related to parts $(P_t^{(1)}, P^{(2)})$}
  if $s \in$ *Data Grid* **then**
    $C' \leftarrow$ *Cell* related to signature s {collision of cells during the merge}
    $\delta c \leftarrow \delta c + \delta c^{(C)}(C, C')$
  end if
end for

best merge among the two variable best merges. The next issue is to efficiently perform the merge and maintain the Data Grid structure.

Performing a merge is similar to evaluating a merge, except that all the components impacted by the merge in the data grid have to be updated. This is detailed in Algorithm 5.

During the completion of a merge, only the cells of the source part have an impact on the time complexity, since they have to be either merged with colliding target cells, either transfered to the target part. The target cells that do not collide with source cells are never considered in the merge completion subroutine. A close inspection of Algorithm 5 confirms that the time complexity of one merge completion is linear in the number of cells in the source part of the merge.

The time complexity of all the merge completions is then related to the total number of cells considered (cell merges or cell transfers) during the whole execution of the GBUM algorithm. Propositions 1 and 2 demonstrate that this process has an overall time complexity of $\mathrm{O}(N \log N)$.

Proposition 1. *The total number of cell merges during the whole execution of the GBUM algorithm is bounded by* N*.*

Proof. Since the initial number of non-empty cells is bounded by N, and since this number is decremented after each cell merge, the total number of cell merges is bounded by N. □

Proposition 2. *The total number of cell transfers or merges during the whole execution of the GBUM algorithm is bounded by* $O(N \log N)$*.*

Proof. Let us focus on one variable and on the part merges related to this variable. Let us first define precisely how we choose the source and target part of a

Algorithm 5 Subroutine: completion of a merge in the core Data Grid structure

Require: *Merge* $M^{(1)}$
{We assume that the inner variable of the merge is variable X_1 and that the number of cells is less in its source part than in its target part}
Ensure: *Data Grid* updated according to the merge
$P_s^{(1)} \leftarrow$ source part of $M^{(1)}$
$P_t^{(1)} \leftarrow$ target part of $M^{(1)}$
for all *Cell* $C \in P_s^{(1)}$ **do**
  $P^{(2)} \leftarrow$ outer part of C {the signature of C is $(P_s^{(1)}, P^{(2)})$}
  Unreference *Cell* C from its source *Part* $P_s^{(1)}$
  Unreference *Cell* C from the *Data Grid*
  $s \leftarrow (P_t^{(1)}, P^{(2)})$ {compute the signature s of C' related to parts $(P_t^{(1)}, P^{(2)})$}
  if $s \in$ *Data Grid* **then**
    {the source *Cell* C is merged in the target *Cell* C'}
    $C' \leftarrow$ *Cell* related to signature s
    Update the output value frequencies of C' with those of C
    Unreference the *Cell* C from its outer *Part* $P^{(2)}$
    Delete the orphan source *Cell* C
  else
    {the source *Cell* C is transfered to the target *Part* $P_t^{(1)}$}
    Replace the *Part* $P_s^{(1)}$ by *Part* $P_t^{(1)}$ in the *Cell* C
    Reference the *Cell* C in its new *Parts* $P_t^{(1)}$
    Reference the *Cell* C in the *Data Grid* with its new signature
  end if
end for
Unreference the source *Part* $P_s^{(1)}$ from the inner *Variable* X_1
Delete the orphan *Part* $P_s^{(1)}$

merge. The part of the merge having the smallest *cell* number is called the *cell-based* source part, and the one having the smallest *instance* number is called the *instance-based* source part. In Algorithm 5, source parts are chosen according to the cell-based rule. In this proof, we first study the algorithmic complexity for a variant of Algorithm 5 using the instance-based instead of the cell-based rule.

Let D be an instance and $\{M_k, 1 \leq k \leq K\}$ the list of the merges in which the instance D is considered, that is where D belongs to the source part. Let n_k be the number of instances in the source part of merge M_k. Since the number of instances in the target part is greater or equal than n_k, the number of instances in the new part resulting from the merge completion is at least twice n_k. The next merge M_{k+1} contains all the instances of this new part. We thus have $n_{k+1} \geq 2n_k$ and more generally $n_k \leq 2^{k-1}n_1$. Since $n_K \leq N$, the number K of merges that consider the instance D is bounded by $\mathrm{O}(\log N)$.

This is true for each instance and for each input variable. Thus, the total number of instances considered in the whole execution of the GBUM algorithm is bounded by $\mathrm{O}(N \log N)$, provided that the instance-based rule is used.

Let us now evaluate the algorithmic complexity using the initial Algorithm 5 which exploits the cell-based rule. The initial algorithm and its variant differ only on the way they choose the source part of each merge. Since the smallest number of cells between two parts is always lower than the smallest number of instances, the total number of cell operations (transfers or merges) is smaller using the cell-based rule of Algorithm 5. The claim follows. □

4.4 Completion of the merges: impacts on the Data Grid Merger

The last maintenance operation, summarized in Algorithm 6, is related to the impact of a merge completion on the other merges. To reach an efficient time complexity, the main issue is to restrict as much as possible the number of inner or outer merges that need to be reevaluated.

Let us analyze the detailed impacts of a merge completion and introduce the principles exploited to tackle the time complexity issue. Figure 4 illustrates the case of a merge in the Wine dataset. Once the merge is completed, at most two inner merges need to be reevaluated: these are the merges adjacent to the source part $P_s^{(1)}$ or to the target part $P_t^{(1)}$ of the merge under completion. On the other hand, potentially all the outer merges (related to variable V_7) have to be reevaluated. In fact, they need to be reevaluated only if they contain at least one cell involved in the merge under completion. More precisely, owing to the additivity of the criterion, the reevaluated outer merges are impacted by at most four cells, which they share with the merge under completion. This is formalized in proposition 3.

Proposition 3. *The number of cells that have an impact on the reevaluation of an outer merge is at most four.*

Proof. Let M be a data grid model with I_1 parts for variable X_1 and I_2 parts for variable X_2. Let $P_{i_{1_s}}^{(1)}$ and $P_{i_{1_t}}^{(1)}$ two parts of variable X_1 involved as the source

Algorithm 6 Subroutine: completion of a merge in the Data Grid Merger structure

Require: *Merge* $M^{(1)}$
{We assume that the inner variable of the merge is variable X_1 and that the number of cells is less in its source part than in its target part}
Ensure: *Data Grid Merger* updated according to the merge
1: **for all** inner merge M that need to be reevaluated **do**
2: Reevaluate M
3: Maintain the sortable list of merges of the inner variable
4: **end for**
5: **for all** outer merge M that need to be reevaluated **do**
6: Reevaluate M
7: Maintain the sortable list of merges of the outer variable
8: **end for**
9: Reevaluate the cost variation $\delta c^{(V)}$ for variable X_1
10: Reevaluate the cost variation $\delta c^{(G)}$ for the data grid

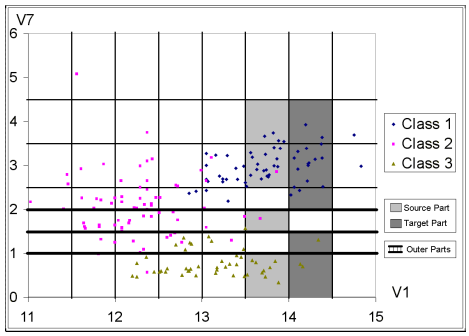


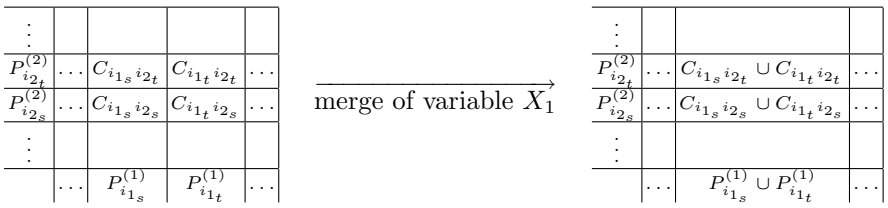
Fig. 4. Merge between the source part $P_s^{(1)} = [13.5, 14.0[$ and the target part $P_t^{(1)} = [14.0, 14.5[$ of the inner variable V_1 in the Wine dataset. One outer merge related to the outer variable V_7 is also represented, with its source part $P_s^{(2)} = [1.0, 1.5[$ and target part $P_t^{(2)} = [1.5, 2.0[$. Four cells (two of them are non-empty cells) shared by the merge under completion and the outer merge have an impact on the reevaluation of the outer merge

and target parts of a merge under completion. Let $P^{(2)}_{i_{2_s}}$ and $P^{(2)}_{i_{2_t}}$ the source and target parts of a merge of the outer variable X_2, M'_2 and M''_2 the data grid resulting from the merge of these outer parts before and after the merge between $P^{(1)}_{i_{1_s}}$ and $P^{(1)}_{i_{1_t}}$.

From the variation of the data grid cost given in formula 4, we get

$$\begin{aligned}\delta c\left(M'_2, M\right) = {} & \delta c^{(G)}\left(I_1 I_2, I_2\right) + \delta c^{(V)}\left(X_2, I_2\right) \\ & + \delta c^{(P)}\left(P^{(2)}_{i_{2_s}}, P^{(2)}_{i_{2_t}}\right) + \sum_{i_1=1}^{I_1} \delta c^{(C)}\left(C_{i_1 i_{2_s}}, C_{i_1 i_{2_t}}\right).\end{aligned} \tag{5}$$

After the merge completion, the new cost variation $\delta c\left(M''_2, M\right)$ involves the same cells belonging to the outer parts $P^{(2)}_{i_{2_s}}$ and $P^{(2)}_{i_{2_t}}$, except those belonging to the parts $P^{(1)}_{i_{1_s}}$ and $P^{(1)}_{i_{1_t}}$. This is illustrated below.



The new cost variation for the outer merge can thus be computed from the former cost variation.

$$\begin{aligned}\delta c\left(M''_2, M\right) = {} & \delta c\left(M'_2, M\right) \\ & + \delta c^{(G)}\left((I_1 - 1) I_2, I_2\right) - \delta c^{(G)}\left(I_1 I_2, I_2\right) \\ & + \delta c^{(C)}\left(C_{i_{1_s} i_{2_s}} \cup C_{i_{1_t} i_{2_s}}, C_{i_{1_s} i_{2_t}} \cup C_{i_{1_t} i_{2_t}}\right) \\ & - \delta c^{(C)}\left(C_{i_{1_s} i_{2_s}}, C_{i_{1_s} i_{2_t}}\right) - \delta c^{(C)}\left(C_{i_{1_t} i_{2_s}}, C_{i_{1_t} i_{2_t}}\right).\end{aligned} \tag{6}$$

The claim follows. □

Sixteen combinations of the four impacted cells may occur, according to whether they are empty or not. Let us call them *cell patterns* and analyze their impact on the cost variation of the outer merge (see formula 6).

The *empty* pattern is $\left|\begin{smallmatrix}\circ & \circ \\ \circ & \circ\end{smallmatrix}\right|$.

The *singleton* pattern ($\left|\begin{smallmatrix}\circ & \circ \\ \circ & \bullet\end{smallmatrix}\right|$, $\left|\begin{smallmatrix}\circ & \circ \\ \bullet & \circ\end{smallmatrix}\right|$, $\left|\begin{smallmatrix}\circ & \bullet \\ \circ & \circ\end{smallmatrix}\right|$, $\left|\begin{smallmatrix}\bullet & \circ \\ \circ & \circ\end{smallmatrix}\right|$) involves one single cell in the outer merge. The cost variation of the outer merge remains unchanged.

The *outer collision* pattern ($\left|\begin{smallmatrix}\bullet & \circ \\ \bullet & \circ\end{smallmatrix}\right|$, $\left|\begin{smallmatrix}\circ & \bullet \\ \circ & \bullet\end{smallmatrix}\right|$) involves exactly two cells that collide for the outer variable. The cost variation of the outer merge remains unchanged since the collision between the two cells is the same before and after the inner merge.

The *inner collision* pattern ($\left|\begin{smallmatrix}\circ&\circ\\ \bullet&\bullet\end{smallmatrix}\right|$, $\left|\begin{smallmatrix}\bullet&\bullet\\ \circ&\circ\end{smallmatrix}\right|$, $\left|\begin{smallmatrix}\circ&\bullet\\ \bullet&\bullet\end{smallmatrix}\right|$, $\left|\begin{smallmatrix}\bullet&\circ\\ \bullet&\bullet\end{smallmatrix}\right|$, $\left|\begin{smallmatrix}\bullet&\bullet\\ \circ&\bullet\end{smallmatrix}\right|$, $\left|\begin{smallmatrix}\bullet&\bullet\\ \bullet&\circ\end{smallmatrix}\right|$, $\left|\begin{smallmatrix}\bullet&\bullet\\ \bullet&\bullet\end{smallmatrix}\right|$) involves at least two cells that collide in the inner merge. The cost variation of the outer merge needs to be reevaluated, since some cells are merged during the part merge under completion.

The *diagonal* pattern ($\left|\begin{smallmatrix}\circ&\bullet\\ \bullet&\circ\end{smallmatrix}\right|$, $\left|\begin{smallmatrix}\bullet&\circ\\ \circ&\bullet\end{smallmatrix}\right|$) involves one outer cell merge after the current inner merge completion, whereas no cell needed to be merged before the merge completion. The cost variation of the outer merge must then be reevaluated.

All in all, a close look at all the cell patterns allow to derive the interesting property 3.

Property 3. An outer merge does not need to be reevaluated when the cell pattern is like $\left|\begin{smallmatrix}\circ&*\\ \circ&*\end{smallmatrix}\right|$, that is when it contains only empty cells in the source part of the merge under completion.

We now present in lemmas 1, 2, 3, 4 and 5 intermediate results related to the time complexity of the Data Grid Merger maintenance operations. We conclude in proposition 4 that the time complexity of all these maintenance operations is $\mathrm{O}(N \log N)$.

Lemma 1. *The total number of inner merges reevaluated during the whole execution of the GBUM algorithm is bounded by $O(N)$.*

Proof. Since at most two inner merges need to be reevaluated after each merge completion, and since the number of merge completions is at most N for each input variable, the claim follows. □

Lemma 2. *The total number of cells considered in the inner merges during the whole execution of the GBUM algorithm is bounded by $O(N \log N)$.*

Proof. The principle of the proof is similar to that of Proposition 2.

For each merge, at most two inner merges (*left* and *right*) must be reevaluated. Let us focus on one variable, on the part merges related to this variable and on the reevaluation of the left inner merges.

In Algorithm 6, source parts are chosen according to the cell-based rule (see Proposition 2 for a definition of the cell-based and instance based rules). In this proof, we first study the algorithmic complexity for a variant of Algorithm 6 using the instance-based instead of the cell-based rule.

Let D be an instance and $\{M_k, 1 \leq k \leq K\}$ the list of the reevaluated left inner merges in which the instance D is considered. Let n_k be the number of instances in the source part of merge M_k. Since the number of instances in the target part is greater or equal than n_k, the number of instances in the new part resulting from the reevaluated merge is at least twice n_k. The next reevaluated merge M_{k+1} contains all the instances of this new part. We thus have $n_{k+1} \geq 2n_k$ and more generally $n_k \leq 2^{k-1} n_1$. Since $n_K \leq N$, the number K of reevaluated left inner merges which consider the instance D is bounded by $\mathrm{O}(\log N)$.

This is true for each instance, for each input variable and for each type (*left* and *right*) of impacted inner merge. Thus, the total number of instances considered in the whole maintenance operations of the Data Grid Merger structure is bounded by $O(N \log N)$, provided that the instance-based rule is used.

Let us now evaluate the algorithmic complexity using the initial Algorithm 6 which exploits the cell-based rule. The initial algorithm and its variant differ only on the way they choose the source part of each merge. Since the smallest number of cells between two parts is always lower than the smallest number of instances, the total number of cell operations (transfers or merges) is smaller using the cell-based rule of Algorithm 6. The claim follows.

□

Lemma 3. *The total number of outer merges reevaluated during the whole execution of the GBUM algorithm is bounded by $O(N \log N)$.*

Proof. Exploiting property 3, an outer merge is reevaluated only if it contains a cell considered in the current inner merge. From proposition 2, the total number of cells considered in the whole merge process is bounded by $O(N \log N)$. The claim follows. □

Lemma 4. *The total number of outer merges the reevaluation of which has a non null impact on the data grid cost is bounded by $O(N)$.*

Proof. Let C be a non-empty cell let us focus on the cell pattern $\left|\begin{smallmatrix} * & * \\ \bullet & * \end{smallmatrix}\right|$, where C is in the lower left corner of the cell pattern. When a merge occurs, a new pattern is associated to the cell C. Let us focus on the chain of the cell patterns related to C during to whole merge process.

This chain can be described by 16 possible states (16 cell patterns) and $256 = 16^2$ possible transitions between the states. A close look at the cell patterns shows that the only outer merges which impact the reevaluation are related to the inner collision pattern and to the diagonal pattern. Concerning the inner collision states, the only possible transitions involve at least one cell merge. Concerning the diagonal state, transitions are possible only to inner collision states or to outer collision states, which contain either two cells in the same row or two cells in the same column. A new diagonal state can be encountered in the chain if and only if a cell merge happens.

The total number of states having an impact on the reevaluation is then bounded by the total number of cell merges in the merge process, that is by $O(N)$ according to proposition 1. This is true for each position of the cell in the cell pattern (lower left, lower right, upper left or upper right corner). The claim follows. □

Lemma 5. *The total number of cells considered in the outer merges during the whole execution of the GBUM algorithm is bounded by $O(N \log N)$.*

Proof. Lemma 3 states that the total number of reevaluated outer merges is bounded by $O(N \log N)$ and proposition 3 states that each reevaluation involves at most four cells. The claim follows. □

Proposition 4. *The overall time complexity of the maintenance operations of the Data Grid Merger structure during the whole execution of the GBUM algorithm is bounded by* $O(N \log N)$.

Proof. The maintenance operations for one merge are summarized in Algorithm 6. From lemmas 2 and 5, the cell cost variation $\delta c^{(C)}$ needs to be computed $O(N \log N)$ times. From lemmas 1 and 3, the part cost variation $\delta c^{(P)}$ needs to be computed $O(N \log N)$ times. From lemma 1 and 4, the number of parts merges which have an impact on the cost reevaluation is $\mathrm{O}(N)$. These merges need to be sorted again in the sorted list of part merges related to each input variable. Each sort maintenance can be performed in $\mathrm{O}(\log N)$, using a maintainable sorted list such as an AVL tree [AVL62]. The total time complexity for the maintenance of the sorted list of merges is then $\mathrm{O}(N \log N)$. The cost variation for the data grid $\delta c^{(G)}$ and for the input variables $\delta c^{(V)}$ needs to be reevaluated after each merge, that is at most $\mathrm{O}(N)$ times. Overall, the time complexity of all the maintenance operations of the Data Grid Merger is thus $\mathrm{O}(N \log N)$. □

4.5 Overall algorithmic complexity

We have shown in section 4.1 that the Data Grid structure can be initialized from the dataset in $\mathrm{O}(N \log N)$ time, and in section 4.2 that all the possible part merges can be evaluated and stored in the Data Grid Merger structure in $\mathrm{O}(N \log N)$ time. These structures need to be maintained during the greedy merge process. We have demonstrated in section 4.3 that all the maintenance operations run in $\mathrm{O}(N \log N)$ time for the Data Grid structure, and in section 4.4 that they run in $\mathrm{O}(N \log N)$ time for the Data Grid Merger structure. The memory complexity is $\mathrm{O}(N)$ both for the Data Grid structure and the Data Grid Merger structure.

Whereas a straightforward implementation of the GBUM algorithm requires an $\mathrm{O}(N^4)$ time complexity and an $\mathrm{O}(N^2)$ memory complexity, a carefully optimized implementation runs in $\mathrm{O}(N \log N)$ time with a $\mathrm{O}(N)$ memory requirement. This optimized implementation exploits both the additivity of the evaluation criterion and the sparseness of the data grids.

5 Post-optimization

The greedy heuristic is time efficient, but it may fall into a local optimum. First, the greedy heuristic may stop too soon and produce too many parts for each input variable. Second, the boundaries of the intervals may be sub-optimal since the merge decisions of the greedy heuristic are never rejected. We propose to reuse the post-optimization algorithms described in [Bou06] in the case of univariate discretization.

In a first stage called *exhaustive merge*, the greedy heuristic merge steps are performed without stopping condition until the data grid consists of one single cell. The best encountered data grid is then memorized. This stage allows escaping local minima with several successive merges and needs $\mathrm{O}(N \log N)$ time.

In a second stage called *greedy post-optimization*, a hill-climbing search is performed in the neighborhood of the best data grid. This search alternates the optimization on each input variable. For a given input variable, the univariate partition is frozen, and the other input variable is optimized using the univariate discretization post-optimization algorithm introduced in [Bou06]. This second stage converges very quickly in practice and requires only a few steps.

We summarize the post-optimization of data grids in Algorithm 7.

Algorithm 7 Post-optimization of a Data Grid

Require: M {Initial data grid solution}
Ensure: $M^*; c(M^*) \leq c(M)$ {Final solution with improved cost}
1: $M^* \leftarrow$ call *exhaustive merge* (M)
2: **while** improved **do**
3: {Univariate post-optimization of variable X_1}
4: freeze the univariate partition of variable X_2
5: $M^* \leftarrow$ call *univariate post-optimization* (M^*) for variable X_1
6: {Univariate post-optimization of variable X_2}
7: freeze the univariate partition of variable X_1
8: $M^* \leftarrow$ call *univariate post-optimization* (M^*) for variable X_2
9: **end while**

The univariate post-optimization exploits a neighborhood of a discretization consisting of combinations of interval splits and interval merges:

- a new interval can be added with one split,
- an interval boundary can be moved with one merge combined with one split,
- an interval can be removed with two merges combined with one split.

Owing to proposition 5, the univariate post-optimization algorithms can be reused directly.

Proposition 5. *Let $c(M)$ be an additive evaluation criterion of a data grid model M. Let $c_1(M_1)$ be the univariate evaluation criterion of the univariate partition model M_1 of variable X_1 when the partition of variable X_2 is frozen. Then $c_1(M_1)$ is an additive univariate evaluation criterion.*

Proof. From equation 3, we get

$$\begin{aligned} c(M) &= c^{(G)}(G) + c^{(V)}(X_1, I_1) + c^{(V)}(X_2, I_2) \\ &+ \sum_{i_1=1}^{I_1} c^{(P)}\left(P_{i_1}^{(1)}\right) + \sum_{i_2=1}^{I_2} c^{(P)}\left(P_{i_2}^{(2)}\right) + \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2} c^{(C)}(C_{i_1 i_2}) \end{aligned} \tag{7}$$

Let us freeze the partition of variable X_2. Thus, the number of parts I_2 and the parts $P_{i_2}^{(2)}$ become constants for the problem of optimizing the univariate

partition of X_1. The criterion $c_1(M_1)$ can be formulated as

$$c_1(M_1) = c_1^{(V)}(X_1, I_1) + \sum_{i_1=1}^{I_1} c_1^{(P)}\left(P_{i_1}^{(1)}\right) \tag{8}$$

with

$$c_1^{(V)}(X_1, I_1) = c^{(G)}(X_1X_2) + c^{(V)}(X_1, I_1) + c^{(V)}(X_2, I_2) + \sum_{i_2=1}^{I_2} c^{(P)}\left(P_{i_2}^{(2)}\right) \tag{9}$$

$$c_1^{(P)}\left(P_{i_1}^{(1)}\right) = c^{(P)}\left(P_{i_1}^{(1)}\right) + \sum_{i_2=1}^{I_2} c^{(C)}(C_{i_1i_2}). \tag{10}$$

The claim follows. □

In the univariate case, each evaluation of a part requires O(1) time, and the overall time complexity of the post-optimization algorithm is $O(N \log N)$ as shown in [Bou06]. In the bivariate case, both the variable criterion $c_1^{(V)}$ and the part criterion $c_1^{(P)}$ are more complex to evaluate. However, the sum term in the variable criterion is constant for a given partition of X_2 and can be evaluated once. The part criterion involves a sum of term for the cells of the current X_1 part related to each (frozen) part of X_2. Since the criterion is null on the empty cells, this sum can be evaluated only for non-empty cells. Thus, the overall evaluation of all the neighbor models of the current best bivariate model require $O(N)$ cell evaluations. Similarly to the main GBUM algorithm, the additivity of the criterion and the sparseness of the data grid can be exploited to perform the greedy post-optimization algorithm in $O(N \log N)$ time.

The post-optimization holds two straightforward notable properties. The first one is that post-optimizing can only improve the results of the GBUM algorithm since its comes after. The second one is that in case of input variables whose optimal joint partitioning consists of one single cell, the optimum data grid is necessarily found owing to the exhaustive merge stage of the post-optimization.

6 Meta-heuristic

The GBUM algorithm allows to evaluate $O(N^2)$ data grid models among $O(2^{2N})$ potential data grids in $O(N \log N)$ time. About $2N$ merges are performed along the algorithm but at most N merges involve a cell collision. This means that about half of the merges have no impact on the mixture of the output values in the cells, so that these merge decisions are blind with respect to the distribution of the data and may destroy interesting patterns. The post-optimization heuristic may bring a significant improvement, but it remains sticked to a close neighborhood of the best encountered solution.

Since the GBUM algorithm is time efficient, it is then natural to apply it repeatedly in order to better explore the search space. This is done according to

Algorithm 8 VNS meta-heuristic for data grid optimization

Require: M {Initial data grid solution}
Require: $MaxLevel$ {Optimization level}
Ensure: $M^*, c(M^* \leq c(M)$ {Final solution with improved cost}
1: $L \leftarrow 1$
2: **while** $L \leq MaxLevel$ **do**
3: {Generate a random solution in the neighborhood of M^*}
4: $M'' \leftarrow$ random solution with $(N/\log N)L/MaxLevel$ intervals per input variable
5: $M' \leftarrow M^* \cup M''$
6: {Optimize and evaluate the new solution}
7: $M' \leftarrow$ call Greedy Bottom-Up Merge(M')
8: $M' \leftarrow$ call Post-Optimization(M')
9: **if** $c(M') < c(M^*)$ **then**
10: $M^* \leftarrow M'$
11: $L \leftarrow 1$
12: **else**
13: $L \leftarrow L + 1$
14: **end if**
15: **end while**

the Variable Neighborhood Search (VNS) [HM01], which consists in applying the primary heuristic (GBUM algorithm and post-optimization) to a neighbor of the solution. If the new solution is not better, a bigger neighborhood is considered. Otherwise, the algorithm restarts with the new best solution and a minimal size neighborhood. The process is controlled by the maximum length of the series of growing neighborhoods to explore.

This meta-heuristic is described in Algorithm 8. According to the level of the neighborhood size l, a new solution M' is generated close to the current best solution. A random discretization for each input variable is obtained with the choice of random interval bounds without replacement. For $L = MaxLevel$, we bound the size of the random discretization by $N/\log N$.

The VNS meta-heuristic only requires the number of sizes of neighborhood as a parameter. This can easily be turned into an anytime optimization algorithm, by calling iteratively the VNS algorithm with parameters of increasing size and stopping the optimization only when the allocated time is elapsed. In [Bou07], all the experiments are performed by calling the VNS algorithm three times with successive parameters equal to 1, 2 and 4.

In order to improve the initial solution, we choose to first optimize the univariate partition of each variable (discretization or value grouping) and to build the initial solution from a cross-product of the univariate partitions. Although this cannot help in case a strictly bivariate patterns (such as XOR for example), this might be helpful otherwise.

7 Adaptation to categorical variables

In marketing applications for example, variables such as Country, State, ZipCode, FirstName, ProductID usually hold many different values. Preprocessing these variables is critical to produce efficient classifiers.

In this section, we focus on categorical variables. We first analyze the balance between intensity of optimization and computation time, propose a practical trade-off, and study the impact of this trade-off on the optimization algorithms.

7.1 Optimization efficiency versus computation time

In the categorical case, $O(V^2)$ part merges have to be evaluated at each step of the GBUM algorithm, instead of $O(N)$ in the numerical case (intervals are constrained to be adjacent, not group of values). When $V \ll N$, this has little impact on the algorithmic time complexity. For large enough V, the optimization algorithms have to be adapted to keep a practical time complexity without sacrifying the quality of the solution.

For $V \geq \sqrt{N}$, the overall time complexity of the optimization algorithms may exceed $O(N \log N)$ and even reach $O(N^2 \log N)$ when $V = N$. However, the number of partitions is based on the Bell number in the case of value grouping instead of the binomial coefficient in the case of discretization. For regularized criteria such as those introduced in [Bou07], the regularization term for the partition grows very quickly with the size of the partition, which penalizes large partitions. Furthermore, since no part can be isolated from the others (no adjacency constraints like for discretizations), the size I of the optimal partition is likely to be small w.r.t the number of instances.

In the following, we assume that $I \leq I_{Max} = \sqrt{N}$ and we restrict all the optimization algorithms to work with constrained variables, according to Definition 5. All the Propositions in section 7 relate implicitly to the case of constrained categorical variables.

Definition 5. *A categorical variable is constrained if the size of the partition of its values is bounded by* $I_{Max} = \sqrt{N}$.

According to Proposition 6, the time complexity of the GBUM algorithm is lower bounded by $O(N\sqrt{N} \log N)$. However, we demonstrate in section 7.2 that this lower bound is also an upper bound of the time complexity of the algorithm.

Proposition 6. *The time complexity of the GBUM algorithm cannot be better than* $O(N\sqrt{N} \log N)$.

Proof. The principle of this proof is based on a counterexample, the evaluation of which in the GBUM algorithm requires at least $O(N\sqrt{N} \log N)$ operations.

Let us consider a data grid with two categorical variables, each of which having $\sqrt{N}$ parts. The number of cells is N. Let us assume that all these cells are non-empty cells, which is possible provided that each cell contains exactly one instance.

When a merge is completed on one variable, both the source and target parts of the merge contain $\sqrt{N}$ cells, which all collide. Thus all the N outer merges need to reevaluated and all of them have a potentially non null impact on the merge cost. Thus, these N merges need to be sorted again in the sorted list of the outer variable, with an overall time complexity of $\mathrm{O}(N \log N)$.

The configuration of the merged part is the same as that of its source and target part, since it always contains $\sqrt{N}$ non-empty cells. Such merges may be performed up to $\sqrt{N}$ times if they are related to the same variable. The claim follows. □

7.2 Impact on the greedy bottom-up heuristic

In this section, we study the algorithmic complexity of the GBUM algorithm in the case of constrained categorical variables. We show in Proposition 7 that the memory complexity of the algorithm is still $\mathrm{O}(N)$ and in Propositions 8, 9 and 10 that its time complexity is $\mathrm{O}(N\sqrt{N}\log N)$.

It is noteworthy that the only algorithmic component with $\mathrm{O}(N\sqrt{N}\log N)$ time complexity is the maintenance of the sorted list of merges per variable, as illustrated in the counterexample given in Proposition 6. All the other components of the GBUM algorithm run in $\mathrm{O}(N\sqrt{N})$ time.

Proposition 7. *The memory complexity of the GBUM algorithm is $O(N)$.*

Proof. The memory complexity of the algorithm is that of the Data Grid and Data Grid Merger structures. The Data Grid structure contains two Variables, $\mathrm{O}(N)$ Parts and $\mathrm{O}(N)$ non-empty cells. The Data Grid Merger structure contains $\mathrm{O}(N)$ Merges, since the number of possible merges if $\mathrm{O}(N)$ both for numerical variables and for constrained categorical variables (${I_{Max}}^2 \leq N$). □

Proposition 8. *The time complexity of the initialization of the Data Grid and Data Grid Merger structures is $O(N \log N)$.*

Proof. The initialization of the Data Grid structure is performed according to Algorithm 2 which has a $\mathrm{O}(N \log N)$ time complexity.

Concerning the initialization of the Data Grid Merger structure, we have to evaluate the number of operations at the part level and cell level to initialize all the merges. Since each part and each non-empty cell is involved in $\mathrm{O}(I_{Max})$ merges, and since there are at most $\mathrm{O}(I_{Max})$ parts and $\mathrm{O}(N)$ non-empty cells, the claim follows. □

Proposition 9. *The time complexity of the maintenance operations of the Data Grid structure during the whole execution of the GBUM algorithm is $O(N\sqrt{N})$.*

Proof. Each merge involves at most two parts and $\mathrm{O}(N)$ cells. Since the number of merges necessary to go from the initial data grid to the terminal data grid is bounded by $\mathrm{O}(I_{Max})$, the claim follows. □

Proposition 10. *The time complexity of the maintenance operations of the Data Grid Merger structure during the whole execution of the GBUM algorithm is* $O(N\sqrt{N}\log N)$.

Proof. The time complexity of the maintenance of the Data Grid Merger structure is related to the number of merges that need to be reevaluated and their impact at the part and cell level. The case of numerical variables has been examined in section 4.4. In this proof, we focus on constrained categorical variables.

After each merge completion, $\mathrm{O}(\sqrt{N})$ inner merges and $\mathrm{O}(N)$ outer merges have to be reevaluated. The total number of reevaluated merges is bounded by $\mathrm{O}(N\sqrt{N})$. The variables need to maintain their sorted lists of merges, which requires $\mathrm{O}(\log N)$ operations per merge. Overall, the maintenance operations have a $\mathrm{O}(N\sqrt{N}\log N)$ time complexity at the merge level and part level (each merges involves two parts).

At the cell level, let us first focus on the total number of cells considered in the inner merges. For a given merge M, let P_0 be the merged part and $P_i, 1 \leq i \leq I$ all the other inner parts of the data grid. Let $N^{(C)}(P_i)$ be the number of cells in part i. Let us evaluate the total number of cells $N^{(C)}(M)$ considered in all the reevaluated inner merges. According to Algorithm 6, the cells are considered only when the cell number is less in the source part than in the target part of the reevaluated merge. We have

$$N^{(C)}(M) \leq \sum_{i=1}^{I} \min(N^{(C)}(P_i), N^{(C)}(P_0)),$$
$$N^{(C)}(M) \leq \sum_{i=1}^{I} N^{(C)}(P_i)),$$
$$N^{(C)}(M) \leq N.$$

Since at most N cells are considered after each merge completion and since there are at most $\mathrm{O}(\sqrt{N})$ merges in the GBUM algorithms, the total number of cells considered in the inner merges is bounded by $\mathrm{O}(N\sqrt{N})$.

Let us finally focus on the total number of cells considered in the outer merges. From Proposition 3, at most four cells need to be considered in the reevaluation of each outer merge. For each inner merge completion, $\mathrm{O}(N)$ outer merges between the $\mathrm{O}(\sqrt{N})$ outer parts need to be reevaluated. Since the GBUM algorithm involves at most $\mathrm{O}(\sqrt{N})$ inner merges, the total number of cells considered in the outer merges is bounded by $\mathrm{O}(N\sqrt{N})$.

The claim follows. □

7.3 Impact on the post-optimization and meta-heuristic

In the post-optimization algorithm, we keep the *exhaustive merge* algorithm, and use an univariate post-optimization algorithm for value grouping derived from [Bou05]. This heuristic consists in evaluating every move of a categorical value

from one group to another and performing the moves as soon as they improve the evaluation criterion. This heuristic is applied on a randomized sort of the categorical values and iterated as long as the criterion is improved. This post-optimization requires $O(VI)$ ($\leq O(N\sqrt{N})$) computation time per iteration and converges very quickly in case of optimized data grid, so that we choose not to bound the number of iterations.

In the meta-heuristic algorithm, we generate random solutions containing at most I_{Max} groups of values for the univariate partitions of categorical variables. Unfortunately, these random partitions are likely to be poor initial solutions for the GBUM heuristic, since each part is a random mixture of values. In order to improve these initial solutions, we pre-optimize them by moving the values across the groups, as described in the post-optimization heuristic. In this pre-optimization, we decide to bound the number of iterations (by two in practice) to both get sufficiently "pure" parts in the initial data grid and control the pre-optimization computation time.

7.4 Synthesis

Overall in the case of categorical variables, a compromise between the time complexity and the quality of the solution is necessary as soon as $V \geq \sqrt{N}$. We choose to constrain the partition of categorical variables to contain at most $O(\sqrt{N})$ parts and showed that even in this case, the computational complexity of the GBUM algorithm can reach $O(N\sqrt{N}\log N)$.

However, we demonstrated that the time complexity of the optimization heuristics is no more than $O(N\sqrt{N}\log N)$ in case of categorical variables with numerous values (beyond $\sqrt{N}$) and that their memory complexity is still $O(N)$. Compared to the numerical case, the same algorithms are used: greedy bottom-up heuristic, post-optimization and meta-heuristic. Another algorithmic component, pre-optimization, needs to be employed in order to "clean" the randomized data grids and feed the meta-heuristic with good initial solutions.

Overall, the time complexity of the optimization heuristics is $O(N\log N)$ in the case of two numerical variables and $O(N\sqrt{N}\log N)$ when one or two of the input variables are categorical.

8 Summary

Data grid models have been introduced in [Bou07] to evaluate the correlation between a pair of input variables and an output variable. They rely on the partition of each input variable into a set of parts (intervals or groups of values). The cross-product of the partitions forms a data grid of cells, each of which allows to locally describe the distribution of the output values.

In this paper, we have introduced the concept of additive evaluation criteria for data grids, which can be decomposed hierarchically as a sum of terms at the data grid, variable, part and cell level.

We have studied the standard greedy top-down merge heuristic, whose straightforward implementation runs in $O(N^4)$ time. After introducing specific algorithmic structures to store a data grid model as well as all the possible merges between parts, we have shown that the time complexity of the heuristic can be reduced to $O(N \log N)$. The optimized heuristic mainly takes benefit of the additivity of the evaluation criterion and of the sparseness of the data grids which contain at most $O(N)$ non-empty cells for $O(N^2)$ cells.

In order to tackle the greediness of the heuristic, we have introduced post-optimization heuristics which exploit local neighborhoods around initial solutions, and a meta-heuristic to globally explore the search space according to neighborhoods of varying size.

Finally, the case of categorical input variables required specific adaptations to strike a balance between the time complexity and the quality of the optimization.

Overall, the data grid optimization algorithms run in $O(N \log N)$ for numerical input variables and in $O(N\sqrt{N} \log N)$ when categorical variables with numerous values (beyond $\sqrt{N}$) are involved. The memory requirement is always $O(N)$. Intensive experiments are reported in [Bou07] to evaluate the optimization algorithms presented in this paper .

References

[AVL62] G. Adelson-Velskii and E.M. Landis. An algorithm for the organization of information. *Doklady Akademii Nauk SSSR, 146 263-266, 1962 (Russian)*, 3:1259–1263, 1962. English translation by Myron J. Ricci in Soviet Math. Doklady.

[BM96] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1996. http://www.ics.uci.edu/mlearn/MLRepository.html.

[Bou04] M. Boullé. Khiops: a statistical discretization method of continuous attributes. *Machine Learning*, 55(1):53–69, 2004.

[Bou05] M. Boullé. A Bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research*, 6:1431–1452, 2005.

[Bou06] M. Boullé. MODL: a Bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1):131–165, 2006.

[Bou07] M. Boullé. Optimal bivariate evaluation for supervised learning using data grid models. *Advances in Data Analysis and Classification*, 2007. submitted.

[Cat91] J. Catlett. On changing continuous attributes into ordered discrete attributes. In *Proceedings of the European Working Session on Learning*, pages 87–102. Springer, 1991.

[DKS95] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning*, pages 194–202. Morgan Kaufmann, San Francisco, CA, 1995.

[ER96] T. Elomaa and J. Rousu. Finding optimal multi-splits for numerical attributes in decision tree learning. Technical Report NC-TR-96-041, Royal Holloway, University of London, 1996. NeuroCOLT.

[FI92] U. Fayyad and K. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8:87–102, 1992.

[HM01] P. Hansen and N. Mladenovic. Variable neighborhood search: principles and applications. *European Journal of Operational Research*, 130:449–467, 2001.

[Hol93] R.C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–90, 1993.

[Kas80] G.V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29(2):119–127, 1980.

[KBR84] I. Kononenko, I. Bratko, and E. Roskar. Experiments in automatic learning of medical diagnostic rules. Technical report, Ljubljana: Joseph Stefan Institute, Faculty of Electrical Engineering and Computer Science, 1984.

[Ker91] R. Kerber. Chimerge discretization of numeric attributes. In *Proceedings of the 10th International Conference on Artificial Intelligence*, pages 123–128. AAAI Press, 1991.

[LHTD02] H. Liu, F. Hussain, C.L. Tan, and M. Dash. Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, 4(6):393–423, 2002.

[Qui93] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[TKS95] T.Fulton, S. Kasif, and S. Salzberg. Efficient algorithms for finding multiway splits for decision trees. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 244–251. Morgan Kaufmann, 1995.

[ZR00] D.A. Zighed and R. Rakotomalala. *Graphes d'induction*. Hermes, France, 2000.

[ZRR98] D.A. Zighed, S. Rabaseda, and R. Rakotomalala. Fusinter: a method for discretization of continuous attributes for supervised learning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(33):307–326, 1998.

E

Segmentation d'image couleur par grille de rectangles optimale selon une approche Bayesienne

Segmentation d'image couleur par grille de rectangles optimale selon une approche Bayesienne

Marc Boullé, Aymeric Larrue

France Télécom R&D
2, avenue Pierre Marzin, 23300 Lannion, France
marc.boulle@orange-ftgroup.com

Résumé − Ce papier introduit une nouvelle méthode de segmentation d'image selon une grille de régions rectangulaires, distribuées de façon homogène sur un ensemble de plages de niveaux de couleurs. La segmentation est ici considérée comme un problème de sélection de modèle, ce qui permet d'établir un critère d'évaluation d'une segmentation optimale selon une approche Bayesienne. Une heuristique d'optimisation performante est également proposée afin de rechercher la meilleure segmentation. Des premières expérimentations démontrent la validité de l'approche et ouvrent des perspectives pour la méthode proposée, en tant que technique générique de prétraitement des images.

Abstract − This paper introduces a new image segmentation method according to a grid of rectangular regions, distributed on a set of color level ranges. The segmentation is considered as a problem of model selection, which enables to establish an optimal evaluation criterion based on a Bayesian approach. An efficient search heuristic is proposed as well, in order to retrieve an efficient image segmentation. First experiments demonstrate the validity of the approach and suggest its potential as a generic technique for image preprocessing.

1 Introduction

La segmentation au sens colorimétrique d'une image consiste à la prétraiter en la partitionnant de façon fiable en zones homogènes vis-à-vis de composantes colorimétriques données. Ce prétraitement est très important pour de nombreuses applications de l'image, qui lors de traitements ultérieurs nécessitent de pouvoir travailler sur des informations fiables et pertinentes. De nombreuses approches ont été proposées pour la segmentation d'image. On distingue essentiellement les approches à base de contours et les approches à base de région [1]. Les approches à base de régions partitionnent les pixels en regroupant les pixels adjacents présentant des composantes colorimétriques similaires. Les régions adjacentes sont regroupées selon un critère prenant en compte un compromis entre l'homogénéité des régions et la simplicité des frontières des régions. Notre approche s'apparente à une approche par région.

Dans ce papier, nous proposons une approche générique basée sur une segmentation d'image en grille de rectangles, chaque rectangle étant caractérisé par la distribution de ses pixels sur des plages de niveaux de couleurs. Les dimensions de la grille sont données par une discrétisation des coordonnées d'espace, et les plages de niveaux résultent d'une discrétisation des couleurs de l'image. Chaque rectangle de la grille doit être le plus homogène possible vis-à-vis des couleurs utilisées, et les rectangles doivent être le plus différent possible (vis-à-vis des couleurs) deux à deux. L'approche s'apparente ainsi à une approche à base de régions, pour laquelle la forme des régions serait contrainte à une grille de rectangles. Le problème principal est alors de trouver un critère d'évaluation d'une segmentation réalisant un compromis efficace entre l'homogénéité des régions (composantes colorimétriques uniformes par rectangle) et la complexité du modèle de segmentation (dimensions de la grille de rectangles et nombre de plages de niveaux par couleur). On propose un critère d'évaluation permettant de trouver la segmentation optimale d'une image en grille de rectangles, résultant de la mise en oeuvre d'une approche Bayesienne de la sélection de modèle.

D'autres travaux apparentés s'appuient sur des segmentations rectangulaires, selon des frontières parallèles aux axes. Par exemple, les segmentations par quadarbre [2] découpent récursivement une image selon les axes horizontaux et verticaux en optimisant un critère d'homogénéité pour les quatre régions ainsi délimitées. En phase de post-traitement, les régions adjacentes similaires sont fusionnées. Par rapport à ces approches, notre méthode recherche une partition en rectangles factorisable sur les deux axes, ce qui fournit un tableau bidimensionnel de régions rectangulaires permettant d'une part une optimisation poussée du critère de segmentation considéré globalement, d'autre part une manipulation aisée des régions lors des traitements ultérieurs.

Dans les approches apparentées, un critère de similitude entre les régions est utilisé afin de décider si deux régions doivent être fusionnées ou rester séparées. Ce critère de similitude nécessite généralement l'ajustement de paramètres, ce qui couplé avec son utilisation répétée pour des décisions locales de fusion ou séparation de régions à différentes échelles, peut entraîner des problèmes de fiabilité de la segmentation obtenue. Notre approche utilise une évaluation MAP globale et sans paramètre de la segmentation, ce qui lui confère à la fois robustesse et simplicité d'utilisation.

Concernant le critère d'évaluation d'une segmentation, d'autres méthodes sont également fondées sur des approches statistiques ou ce qui est équivalent sur des approches de type MDL [3]. Ces approches se basent généralement sur des hypothèses paramétriques de distribution des couleurs, Gaussienne [4] ou Poissonnienne [5] dans le cas des images radar par exemple. Notre approche n'exploite que les connaissances génériques des images, dimensions et nombres de nivaux par couleurs. Elle ne fait d'autre hypothèse que celle d'une distribution uniforme des plages de nivaux de couleurs par région rectangulaire et des nivaux de couleur par plage de nivaux.

2 Modèle d'image couleur en grille

Après avoir présenté notre représentation des images couleur, on introduit les modèles de segmentation en grille. On décrit l'approche Bayesienne permettant d'évaluer la qualité de tels modèles et on présente l'algorithme d'optimisation qui permet de rechercher la meilleure segmentation.

2.1 Représentation d'une image couleur

On considère une image couleur comme un tableau de données bi-dimensionnel dont les instances sont les pixels de l'image et les variables, au nombre de cinq, sont deux variables d'espace et trois variables colorimétriques. Les variables colorimétriques peuvent être (R, G, B), (H, S, V), ou même se réduire à une seule variable de niveaux de gris par exemple. Dans le reste du papier, pour des raisons de commodité dans les notations, on se basera sur l'espace colorimétrique (R, G, B).

Soit $\mathbb{V} = \{X, Y, R, G, B\}$ l'ensemble des variables d'une image, se décomposant en un sous ensemble $\mathbb{V}_S = \{X, Y\}$ de variables d'espace et un sous-ensemble $\mathbb{V}_C = \{R, G, B\}$ de variables de couleur. En utilisant la connaissance a priori du domaine des images et de leur codage au format bitmap, on sait que chacune des variables peut prendre un nombre fini de valeurs entières, selon les constantes suivantes : M_X, M_Y : largeur et hauteur en pixels de l'image, $N = M_X M_Y$: nombre de pixels, M_R, M_G, M_B : nombres de nivaux pour les variables de couleurs R, G, B.

2.2 Modélisation d'une image couleur

On définit un modèle d'image comme un modèle statistique de régression des couleurs, plus précisément comme un modèle d'estimation de la densité des variables colorimétriques conditionnellement aux variables d'espace.

De façon similaire à l'objectif classique des méthodes de segmentation d'image, on recherche des régions d'espace homogènes vis à vis de la distribution des couleurs. Pour définir les régions, on discrétise chaque variable d'espace en un ensemble d'intervalles adjacents. Le produit cartésien des discrétisations des variables X et Y définit un ensemble de régions rectangulaires, sur lesquelles on définit la distribution des couleurs. Le nombre de niveaux étant important, on discrétise également chaque variable de couleur en plages de niveaux adjacentes, en supposant que la répartition des niveaux est homogène localement à chaque plage de niveaux. Pour chaque rectangle d'espace, on décrit la distribution des pixels sur les plages de niveaux. L'ensemble des rectangles d'espace, des plages de niveaux, et des distributions des pixels de chaque rectangle sur les plages de niveaux fournit alors un modèle synthétique de la densité des composantes colorimétriques conditionnellement à la position des pixels.

Cette approche est formalisée dans la définition 1.

Définition 1 *Un modèle d'image en grille est défini par la discrétisation de chaque variable d'espace et de couleur et sur chaque région rectangulaire de la partition de l'image en grille ainsi définie par la distribution des pixels sur les plages de niveaux de couleur.*

Les paramètres d'un modèle d'image en grille sont entièrement définis par :

- I_X, I_Y : nombre d'intervalles de largeur et de hauteur,
- I_R, I_G, I_B : nombre de plages de niveaux pour R, G, B,
- N_x^X : largeur en pixels de l'intervalle x de X,
- N_y^Y : hauteur en pixels de l'intervalle y de Y,
- N_i^V : nombre de niveaux pour la plage i de la variable de couleur V ($V \in \mathbb{V}_C$).
- N_{xyi}^V : nombre de pixels de la région (x, y) pour la plage de niveaux i de la couleur V.

Soient enfin N_{xy} le nombre de pixels de la région (x, y) et $N_i^{[V]}$ le nombre total de pixels de l'image pour la plage de niveaux i de la variable de couleur V. Ces quantités sont déduites de la connaissance du format bitmap des images et des paramètres d'un modèle d'image en grille selon les équations $N_{xy} = N_x^X N_y^Y$ et $N_i^{[V]} = \sum_{x=1}^{I_X} \sum_{y=1}^{I_Y} N_{xyi}^V$.

2.3 Evaluation d'un modèle en grille

Les modèles d'image en grille sont très expressifs, puisqu'ils peuvent passer d'une description très grossière de l'image en une seule région et une seule plage de couleur (modèle nul) à une description très fine contenant autant de régions élémentaires que de pixels et autant de plages que de niveaux potentiels (modèle complet).

Nous proposons ici de formuler le choix de la meilleure granularité d'un modèle en grille comme un problème de sélection de modèle. Une approche Bayesienne est appliquée pour choisir le meilleur modèle en grille, qui est recherché en maximisant la probabilité p(Modèle|Données) du modèle en grille sachant les données, c'est à dire l'image. En utilisant la règle de Bayes, et puisque la probabilité des données p(Données) ne dépend pas du modèle, il s'agit alors de maximiser p(Modèle)p(Données|Modèle), c'est-à-dire le produit d'un terme d'a priori sur les modèles et d'un terme de vraisemblance des données connaissant le modèle.

On propose une distribution a priori des modèles en grille exploitant la hiérarchie des paramètres : les nombres d'intervalles sont d'abord choisis sur chaque variable d'espace et de couleur, puis les bornes des intervalles et enfin les effectifs par plage de couleur dans chaque région. Le choix est uniforme à chaque étage de cette hiérarchie. De plus, les distributions des plages de niveaux de couleurs par région sont supposées indépendantes entre elles.

En utilisant la définition de la famille de modèles en grille et sa distribution a priori, la formule de Bayes permet de calculer explicitement les probabilités a posteriori des modèles connaissant les données. En prenant le log négatif de ces probabilités, cela conduit au critère d'évaluation d'un modèle en grille M fourni dans l'équation 1 :

$$\begin{aligned} C(M) = & \sum_{V\in\mathbb{V}} \log M_V \\ & + \sum_{V\in\mathbb{V}} \log \binom{M_V + I_V - 1}{I_V - 1} \\ & + \sum_{x=1}^{I_X}\sum_{y=1}^{I_Y}\sum_{V\in\mathbb{V}_C} \log \binom{N_{xy} + I_V - 1}{I_V - 1} \\ & + \sum_{x=1}^{I_X}\sum_{y=1}^{I_Y}\sum_{V\in\mathbb{V}_C} \left(\log N_{xy}! - \sum_{i=1}^{I_V} N_{xyi}^V! \right) \\ & + \sum_{V\in\mathbb{V}_C}\sum_{i=1}^{I_V} N_i^{[V]} \log N_i^V \end{aligned} \quad (1)$$

La première ligne de l'équation 1 correspond au choix des nombres d'intervalles, la deuxième ligne au choix des bornes de ces intervalles. La troisième ligne correspond au choix des paramètres de la distribution multinomiale des pixels de chaque région sur les plages de niveaux de couleurs, indépendamment pour chacune des variables de couleur. La quatrième ligne est un terme du multinôme pour chaque région, représentant la vraisemblance d'observer les pixels distribués suivant les paramètres de la multinomiale définie sur la troisième ligne. Le terme de la dernière ligne se base sur le nombre de niveaux possibles N_i^V de chaque plage et sur l'effectif cumulé $N_i^{[V]}$ des pixels de l'image sur cette plage. Il s'agit également d'un terme de vraisemblance du niveau des pixels sur leur plage, en se basant sur une hypothèse de distribution uniforme des niveaux dans chaque plage.

Il est à noter que comme le log négatif d'une probabilité représente une longueur de codage [6], la formule 1 s'interprète comme une longueur d'encodage d'une image, avec la partie d'a priori encodant les paramètres du modèle en grille et la partie de vraisemblance encodant la position et le niveau colorimétrique exact des pixels connaissant le modèle d'image. On peut vérifier que les deux modèles "extrêmes" (nul et complet) ont une longueur de codage approximativement égale à celle du format bitmap natif, qui utilise 24 bits par pixel. Les modèle intermédiaires sont à même d'identifier des régions homogènes vis à vis des couleurs, ce qui leur permet un codage plus efficace.

2.4 Optimisation d'une segmentation

On résume dans l'algorithme 1 une heuristique d'optimisation consistant à optimiser alternativement la discrétisation de chaque variable de l'image, en figeant la discrétisation des autres variables. L'algorithme de discrétisation utilisé est celui présenté dans [7] dans le cadre de la discrétisation des variables numériques pour les problèmes de classification supervisée du Data Mining. Les optimisations sont réitérées tant qu'il y a amélioration du critère d'évaluation. En pratique, comme le montre la figure 1, l'optimum est vite atteint indépendamment du modèle initial en entrée de l'algorithme, en moins de trois itérations sur l'ensemble des variables selon nos expérimentations.

Algorithme 1 Optimisation d'une image en grille

Entrée: M {Modèle initial}
Sortie: $M^*; c(M^*) \leq c(M)$ {Modèle final amélioré}
1: **Tant que** amélioration **répéter**
2: **Pour tout** $V \in \mathbb{V}$ **répéter**
3: Figer la discrétisation des autres variables
4: Optimiser la discrétisation de la variable V
5: **Fin pour**
6: **Fin tant que**

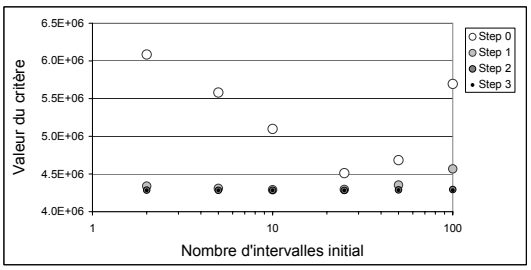


FIG. 1 – Valeur du critère après 0 à 3 étapes d'optimisation, en partant de modèles initiaux pour 2, 5, 10, 25, 50 et 100 intervalles équidistants par variable.

3 Evaluation

On présente dans cette section des résultats d'expérimentations préliminaires sur la segmentation d'une image en grille et ses applications potentielles pour la recherche de zone d'intérêt, la compression d'image ou la détection de mouvement.

3.1 Exemple de segmentation en grille

La méthode de segmentation en grille est évaluée en utilisant l'image Lena de dimension $512 * 512$ au format RGB, dont le fichier bitmap a une taille de 768 ko.

La segmentation obtenue est représentée sur la gauche de la figure 2. Cette grille comprend 57 intervalles horizontaux et 27 intervalles verticaux, soient 1539 régions à comparer avec les 262144 pixels de la représentation initiale. Les couleurs sont discrétisées en 21 plages pour le rouge, 25 pour le vert et 20 pour le bleu. Dans la droite de la figure 2, les régions de la grille sont dessinées en prenant leur couleur moyenne, ce qui permet de définir un modèle simplifié de l'image de taille inférieure à 5 ko, soit environ 0.5% de la taille de la bitmap. Ce format est très aisément manipulable puisque les régions rectangulaires sont indexables directement dans un tableau bidimensionnel. Les motifs de l'image sont préservés à un niveau macroscopique, comme le montre la figure 2. La segmentation en grille s'avère ainsi potentiellement intéressante en tant que prétraitement pour la recherche de zones d'intérêt.

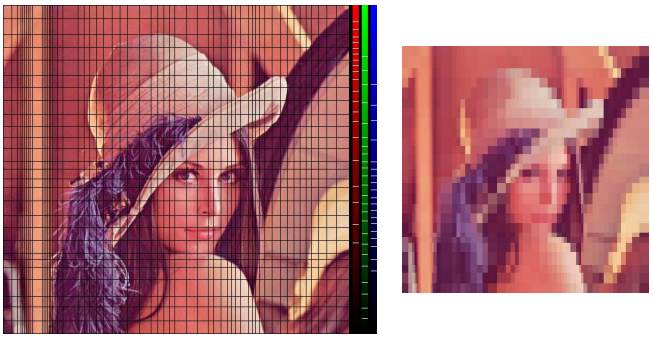


FIG. 2 – Grille de la segmentation et plages de niveaux de couleur pour l'image Lena ; à droite, le modèle en grille avec couleur moyenne par région.

3.2 Application à la compression d'image

La valeur du critère d'évaluation de la grille pour l'image Lena est 4281750 en prenant la base 2 du logarithme dans la formule (1). Selon la section 2.3, cette valeur correspond au coût de codage théorique de l'image. Ce coût de codage permettrait en théorie de comprimer l'image Lena sans perte d'information en utilisant environ 69% de la taille de l'image au format bitmap initial. A titre de coût de comparaison, le format PNG permet une compression sans perte pour un facteur d'environ 78%.

Ce résultat préliminaire permet d'envisager l'utilisation des modèles d'image en grille pour la compression d'image. Il montre surtout que la démarche Bayesienne de choix de la granularité de la grille conduit à une solution particulièrement performante : le compromis est optimal entre la complexité de la grille et la fidélité de la description des régularités de l'image.

3.3 Application à la détection de mouvement

On propose ici d'évaluer l'utilisation potentielle des modèles d'image en grille pour la détection de mouvement. Dans un flux vidéo, on effectue la différence entre deux trames successives afin de détecter les mouvements en se basant sur les régions ayant changé de couleur. La figure 3 présente un tel exemple dans le cadre d'un flux video en niveaux de gris.

FIG. 3 – Détection de mouvement dans un flux vidéo : à gauche, image résultant de la différence entre deux trames successives ; à droite, segmentation en grille de cette image avec visualisation des zones de forte entropie

Le traitement de ce type d'image pose le problème de l'identification des zones de mouvement, à distinguer des zones de bruits vidéo liée par exemple aux variations locales de luminosité ou aux caractéristiques des capteurs de la caméra. Du fait de la robustesse de l'approche Bayesienne utilisée, la technique de segmentation en grille est appropriée pour le prétraitement de ce type d'image. Dans le cas de l'image présentée sur la gauche de la figure 3, la grille obtenue comporte $72 = 13 * 4$ régions distribuées sur 6 plages de niveaux de gris. La droite de la figure 3 représente la grille obtenue en coloriant chaque région selon son entropie. Les régions d'entropie nulle ou faible (en blanc ou gris clair) sont celles où l'image est restée constante ou a légèrement varié en raison du bruit vidéo. Les régions de forte entropie (gris foncé) correspondent aux zones ayant le plus changé, ce qui permet d'identifier les zones candidates pour la détection de mouvement.

4 Conclusion

Les modèles d'image en grille proposés ici se basent sur un partitionnement de l'image en une grille bidimensionnelle de régions rectangulaires et sur la discrétisation des couleurs en plages. L'apport de la méthode réside principalement dans sa généricité, qui la rend indépendante du domaine d'application, et dans l'approche Bayesienne utilisée pour l'évaluation globale des modèles d'image, qui permet d'obtenir une granularité optimale des grilles de segmentation.

Des expérimentations préliminaires confirment la validité de l'approche et illustrent son potentiel en tant que méthode générique de prétraitement d'image, dans le cadre de la segmentation, de la compression sans perte d'information ou de la détection de mouvement. Les travaux à venir viseront à finaliser l'approche présentée et à développer son potentiel pour le prétraitement d'image.

Références

[1] H. Maître. *Le traitement des images*. Lavoisier, 2003.

[2] S.L. Horowitz and T. Pavlidis. Picture segmentation by a tree traversal algorithm. *JACM*, 23(2) :368–388, April 1976.

[3] J. Rissanen. Modeling by shortest data description. *Automatica*, 14 :465–471, 1978.

[4] U. Ndili, R. Nowak, and M. Figueiredo. Coding theoretic approach to image segmentation. In *IEEE International Conference on Image Processing*, October 2001.

[5] V. Venkatachalam, R.D. Nowak, R.G. Baraniuk, and M.A. Figueiredo. Unsupervised sar image segmentation using recursive partitioning. In E.G. Zelnio, editor, *Proc. SPIE, Algorithms for Synthetic Aperture Radar Imagery VII*, volume 4053, pages 121–129, 2000.

[6] C.E. Shannon. A mathematical theory of communication. Technical report, Bell systems technical journal, 1948.

[7] M. Boullé. MODL : a Bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1) :131–165, 2006.

Bibliographie

[Adelson-Velskii and Landis, 1962] G. Adelson-Velskii and E.M. Landis. An algorithm for the organization of information. *Doklady Akademii Nauk SSSR, 146 263-266, 1962 (Russian)*, 3 :1259–1263, 1962. English translation by Myron J. Ricci in Soviet Math. Doklady.

[Adriaans and Vitányi, 2006] P. Adriaans and P. Vitányi. The power and perils of MDL. *ArXiv Computer Science e-prints*, 2006.

[Agrawal *et al.*, 1993] R. Agrawal, T. Imielinski, and A.N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD conference on management of data*, pages 207–216, Washington, D.C., 1993.

[Agrawal *et al.*, 1998] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 94–105, 1998.

[Bay and Pazzani, 1999] S.D. Bay and M.J. Pazzani. Detecting change in categorical data : Mining contrast sets. In *Knowledge Discovery and Data Mining*, pages 302–306, 1999.

[Bay, 2001] S.D. Bay. Multivariate discretization for set mining. *Machine Learning*, 3(4) :491–512, 2001.

[Bekkerman *et al.*, 2005] R. Bekkerman, R. El-Yaniv, and A. McCallum. Multi-way distributional clustering via pairwise interactions. In *Proceedings of ICML-05, the 22nd International Conference on Machine Learning*, pages 41–48, 2005.

[Berckman, 1995] N. C. Berckman. Value grouping for binary decision trees. Technical report, Computer Science Department, University of Massachusetts, 1995.

[Berger, 2006] J. Berger. The case of objective bayesian analysis. *Bayesian Analysis*, 1(3) :385–402, 2006.

[Bernardo and Smith, 2000] J.M. Bernardo and A.F.M. Smith. *Bayesian theory*. John Wiley & sons, 2000.

[Bertier and Bouroche, 1981] P. Bertier and J.M. Bouroche. *Analyse des données multidimensionnelles*. Presses Universitaires de France, 1981.

[Birgé and Rozenholc, 2002] L. Birgé and Y. Rozenholc. How many bins should be put in a regular histogram. *Prépublication 721, Laboratoire de Probabilités et Modèles Aléatoires, Université Paris VI et VII, France*, 2002.

[Blake and Merz, 1996] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1996. http ://www.ics.uci.edu/mlearn/MLRepository.html.

[Bock, 1979] H. Bock. Simultaneous clustering of objects and variables. In E. Diday, editor, *Analyse des Données et Informatique*, pages 187–203. INRIA, 1979.

[Boullé and Hue, 2006] M. Boullé and C. Hue. Optimal Bayesian 2D-discretization for variable ranking in regression. In *Ninth international conference on discovery science*, pages 53–64, 2006.

[Boullé and Larrue, 2007] M. Boullé and A. Larrue. Segmentation d'image couleur par grille de rectangles optimale selon une approche Bayesienne. In *GRETSI 2007*, 2007.

[Boullé, 2003] M. Boullé. Khiops : a discretization method of continuous attributes with guaranteed resistance to noise. In P. Perner and A. Rosenfeld, editors, *Proceedings of the Third International Conference on Machine Learning and Data Mining in Pattern Recognition*, volume 2734 of *LNAI*, pages 50–64. Springer, 2003.

[Boullé, 2004a] M. Boullé. Khiops : a statistical discretization method of continuous attributes. *Machine Learning*, 55(1) :53–69, 2004.

[Boullé, 2004b] M. Boullé. MODL : une méthode quasi-optimale de discrétisation supervisée. Technical Report 8444, France Telecom R&D, 2004.

[Boullé, 2004c] M. Boullé. A robust method for partitioning the values of categorical attributes. In *Extraction et gestion des connaissances (EGC'2004)*, pages 173–184, 2004.

[Boullé, 2005a] M. Boullé. A Bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research*, 6 :1431–1452, 2005.

[Boullé, 2005b] M. Boullé. A grouping method for categorical attributes having very large number of values. In P. Perner and A. Imiya, editors, *Proceedings of the Fourth International Conference on Machine Learning and Data Mining in Pattern Recognition*, volume 3587 of *LNAI*, pages 228–242. Springer verlag, 2005.

[Boullé, 2005c] M. Boullé. Optimal bin number for equal frequency discretizations in supervized learning. *Journal of intelligent data analysis*, 9(2) :175–188, 2005.

[Boullé, 2006a] M. Boullé. An enhanced selective naive bayes method with optimal discretization. In I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, editors, *Feature Extraction : Foundations And Applications*, chapter 25, pages 499–507. Springer, 2006.

[Boullé, 2006b] M. Boullé. MODL : a Bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1) :131–165, 2006.

[Boullé, 2006c] M. Boullé. Regularization and averaging of the selective naive Bayes classifier. In *International Joint Conference on Neural Networks*, pages 2989–2997, 2006.

[Boullé, 2007a] M. Boullé. Compression-based averaging of selective naive Bayes classifiers. *Journal of Machine Learning Research*, 8 :1659–1685, 2007.

[Boullé, 2007b] M. Boullé. Optimal bivariate evaluation for supervised learning using data grid models. *Advances in Data Analysis and Classification*, 2007. submitted.

[Boullé, 2007c] M. Boullé. Optimization algorithms for bivariate evaluation of data grid models. *Advances in Data Analysis and Classification*, 2007. submitted.

[Boullé, 2007d] M. Boullé. Report on preliminary experiments with data grid models in the agnostic learning vs. prior knowledge challenge. In *Proceedings of International Joint Conference on Neural Networks*, 2007. Paper number 1802.

[Breiman *et al.*, 1984] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees.* California : Wadsworth International, 1984.

[Breiman, 1996] L. Breiman. Bagging predictors. *Machine Learning*, 24(2) :123–140, 1996.

[Breiman, 2001] L. Breiman. Random forests. *Machine Learning*, 45(1) :5–32, 2001.

[Castellan, 1999] G. Castellan. Modified akaike's criterion for histogram density estimation. *Technical Report, Université Paris-Sud, Orsay*, 1999.

[Catlett, 1991] J. Catlett. On changing continuous attributes into ordered discrete attributes. In *Proceedings of the European Working Session on Learning*, pages 87–102. Springer, 1991.

[Cawley *et al.*, 2006] G.C. Cawley, M.R. Haylock, and S.R. Dorling. Predictive uncertainty in environmental modelling. In *International Joint Conference on Neural Networks*, pages 11096–11103, 2006.

[Cestnik *et al.*, 1987] B. Cestnik, I. Kononenko, and I. Bratko. Assistant 86 : A knowledge-elicitation tool for sophisticated users. In Bratko and Lavrac, editors, *Proceedings of the 2nd European Working Session on Learning*, pages 31–45. Sigma Press, 1987.

[Chao and Li, 2005] S. Chao and Y. Li. Multivariate interdependent discretization for continuous attribute. In *Third International Conference on Information Technology and Applications (ICITA'05)*, volume 1, pages 167–172. IEEE Computer Society, 2005.

[Chapelle *et al.*, 2006] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning.* MIT Press, Cambridge, MA, 2006. (in press).

[Chapman *et al.*, 2000] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. *CRISP-DM 1.0 : step-by-step data mining guide*, 2000.

[Chaudhuri and Loh, 2002] P. Chaudhuri and W.-Y. Loh. Nonparametric estimation of conditional quantiles using quantile regression trees. *Bernouilli*, 8 :561–576, 2002.

[Chaudhuri *et al.*, 1994] P. Chaudhuri, M.-C. Huang, W.-Y. Loh, and R. Yao. Piecewise-polynomial regression trees. *Statistica Sinica*, 4 :143–167, 1994. This is the first paper on polynomial regression trees.

[Chlebus and Nguyen, 1998] B.S. Chlebus and S.H. Nguyen. On finding optimal discretizations for two attributes. *Rough Sets and Current Trends in Computing*, pages 537–544, 1998.

[Chou, 1991] P. A. Chou. Optimal partitioning for classification and regression trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4) :340–354, 1991.

[Chu and Ghahramani, 2005] W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6 :1019–1041, 2005.

[Chu and Keerthi, 2005] W. Chu and S. Keerthi. New approaches to support vector ordinal regression. In *In ICML '05 : Proceedings of the 22nd international conference on Machine Learning*, 2005.

[Cochran, 1954] W.G. Cochran. Some methods for strengthening the common chi-squared tests. *Biometrics*, 10(4) :417–451, 1954.

[Connor-Linton, 2003] J. Connor-Linton. Chi square tutorial, 2003. http ://www.georgetown.edu/faculty/ballc/webtools/web_chi_tut.html.

[Crammer and Singer, 2001] K. Crammer and Y. Singer. Pranking with ranking. In *Proceedings of the Fourteenth Annual Conference on Neural Information Processing Systems (NIPS)*, 2001.

[Dhillon *et al.*, 2003] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD-2003)*, pages 89–98, 2003.

[Dom, 1997] B. E. Dom. Mdl estimation for small sample sizes and its application to segmenting binary strings. In *CVPR '97 : Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 280–287. IEEE Computer Society, 1997.

[Dougherty *et al.*, 1995] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning*, pages 194–202. Morgan Kaufmann, San Francisco, CA, 1995.

[El-Yaniv and Souroujon, 2001] R. El-Yaniv and O. Souroujon. Iterative double clustering for unsupervised and semi-supervised learning. In Luc De Raedt and Peter A. Flach, editors, *Proceedings of ECML-01, 12th European Conference on Machine Learning*, pages 121–132. Springer Verlag, Heidelberg, DE, 2001.

[Elomaa and Rousu, 1996] T. Elomaa and J. Rousu. Finding optimal multi-splits for numerical attributes in decision tree learning. Technical Report NC-TR-96-041, Royal Holloway, University of London, 1996. NeuroCOLT.

[Elomaa and Rousu, 1999] T. Elomaa and J. Rousu. General and efficient multisplitting of numerical attributes. *Machine Learning*, 36 :201–244, 1999.

[Elomaa *et al.*, 2005] T. Elomaa, J. Kujala, and J. Rousu. Approximation algorithms for minimizing empirical error by axis-parallel hyperplanes. In *Machine Learning : ECML 2005*, pages 547–555. Springer Verlag, Heidelberg, DE, 2005.

[Fan *et al.*, 1996] J. Fan, Q. Yao, and H. Tong. Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems. *Biometrika*, 83 :189–196, 1996.

[Fawcett, 2003] T. Fawcett. ROC graphs : Notes and practical considerations for researchers. Technical Report HPL-2003-4, HP Laboratories, 2003.

[Fayyad and Irani, 1992] U. Fayyad and K. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8 :87–102, 1992.

[Fayyad *et al.*, 1996] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery : An overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–34. AAAI/MIT Press, 1996.

[Ferrandiz and Boullé, 2006] S. Ferrandiz and M. Boullé. Supervised evaluation of voronoi partitions. *Journal of intelligent data analysis*, 10(3) :269–284, 2006.

[Ferrandiz, 2006] S. Ferrandiz. *Apprentissage supervisé à partir de données séquentielles*. PhD thesis, Université de Caen, 2006.

[Fischer, 1958] W.D. Fischer. On grouping for maximum of homogeneity. *Jounal of the American Statistical Association*, 53 :789–798, 1958.

[Fisher, 1936] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7 :179–188, 1936.

[Frank *et al.*, 1998] E. Frank, L. Trigg, G. Holmes, and I. Witten. Naive Bayes for regression, 1998. Working Paper 98/15. Hamilton, NZ : Waikato University, Department of Computer Science.

[Frey and Slate, 1991] P.W. Frey and D.J. Slate. Letter recognition using holland-style adaptive classifiers. *Machine Learning*, 6(2) :161–182, 1991.

[Friedman and Goldszmidt, 1996] N. Friedman and M. Goldszmidt. Discretizing continuous attributes while learning bayesian networks. In *International Conference on Machine Learning*, pages 157–165, 1996.

[Fulton *et al.*, 1995] T. Fulton, S. Kasif, and S. Salzberg. Efficient algorithms for finding multi-way splits for decision trees. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 244–251. Morgan Kaufmann, 1995.

[Goldstein, 2006] M. Goldstein. Subjective bayesian analysis : principles and practice. *Bayesian Analysis*, 1(3) :403–420, 2006.

[Govaert and Nadif, 2003] G. Govaert and M. Nadif. Clustering with block mixture models. *Pattern Recognition*, 36(2) :463–473, 2003.

[Govaert and Nadif, 2005] G. Govaert and M. Nadif. An em algorithm for the block mixture model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(4) :643–647, 2005.

[Govaert and Nadif, 2006] G. Govaert and M. Nadif. Classification d'un tableau de contingence et modèle probabiliste. *Revue des Nouvelles Technologies de l'Information*, 2 :457–462, 2006.

[Grünwald *et al.*, 2005] P.D. Grünwald, I.J. Myung, and M.A. Pitt. *Advances in minimum description length : theory and applications*. MIT Press, 2005.

[Guyon and Elisseeff, 2003] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3 :1157–1182, 2003.

[Guyon *et al.*, 2006a] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, editors. *Feature Extraction : Foundations And Applications*. Springer, 2006.

[Guyon *et al.*, 2006b] I. Guyon, A.R. Saffari, G. Dror, and J.M. Bumann. Performance prediction challenge. In *International Joint Conference on Neural Networks*, pages 2958–2965, 2006. http ://www.modelselect.inf.ethz.ch/index.php.

[Guyon *et al.*, 2007] I. Guyon, A.R. Saffari, G. Dror, and G. Cawley. Agnostic learning vs. prior knowledge challenge. In *International Joint Conference on Neural Networks*, 2007.

[Guyon, 2003] I. Guyon. Design of experiments of the nips 2003 variable selection benchmark, 2003. http ://www.nipsfsc.ecs.soton.ac.uk/papers/NIPS2003-Datasets.pdf.

[Hand and Yu, 2001] D.J. Hand and K. Yu. Idiot bayes ? not so stupid after all ? *International Statistical Review*, 69(3) :385–399, 2001.

[Hansen and Mladenovic, 2001] P. Hansen and N. Mladenovic. Variable neighborhood search : principles and applications. *European Journal of Operational Research*, 130 :449–467, 2001.

[Hansen and Yu, 2001] M.H. Hansen and B. Yu. Model selection and the principle of minimum description length. *J. American Statistical Association*, 96 :746–774, 2001.

[Hartigan, 1972] J.A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337) :123–129, 1972.

[Herbrich *et al.*, 2000] R. Herbrich, T. Graepel, and K. Obermayer. *Advances in Large Margin Classifiers*, chapter 7, pages 115–132. MIT Press, 2000.

[Hettmansperger and McKean, 1998] T. P. Hettmansperger and J. W. McKean. *Robust Nonparametric Statistical Methods*. Arnold, London, 1998.

[Hoeting *et al.*, 1999] J.A. Hoeting, D. Madigan, A.E. Raftery, and C.T. Volinsky. Bayesian model averaging : A tutorial. *Statistical Science*, 14(4) :382–417, 1999.

[Holte, 1993] R.C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11 :63–90, 1993.

[Hue and Boullé, 2007a] C. Hue and M. Boullé. A new probabilistic approach in rank regression with optimal bayesian partitioning. *Journal of Machine Learning Research*, 2007. Accepted for publication.

[Hue and Boullé, 2007b] C. Hue and M. Boullé. Une approche non paramétrique bayesienne pour l'estimation de densité conditionnelle sur les rangs. In *Extraction et gestion des connaissances (EGC'2007)*, pages 111–122, 2007.

[Jaynes, 2003] E.T. Jaynes. *Probability Theory The Logic of Science*. Cambridge University Press, 2003.

[Joachims, 1997] T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In D.H. Fisher, editor, *Proceedings of 14th International Conference on Machine Learning*, pages 143–151. Morgan Kaufmann Publishers, San Francisco, US, 1997.

[Kass, 1980] G.V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29(2) :119–127, 1980.

[Kerber, 1991] R. Kerber. Chimerge discretization of numeric attributes. In *Proceedings of the 10th International Conference on Artificial Intelligence*, pages 123–128. AAAI Press, 1991.

[Kira and Rendell, 1992] K. Kira and L.A. Rendell. A practical approach to feature selection. In *ML92 : Proceedings of the ninth international workshop on Machine learning*, pages 249–256. Morgan Kaufmann Publishers Inc., 1992.

[Knuth, 1997] D. Knuth. *The Art of Computer Programming, Volume 3 : Sorting and Searching*. Addison-Wesley, New York, 1997. Third Edition.

[Koenker, 2005] R. Koenker. *Quantile Regression*. Econometric Society Monograph Series. Cambridge University Press, 2005.

[Kohavi and John, 1997] R. Kohavi and G. John. Wrappers for feature selection. *Artificial Intelligence*, 97(1-2) :273–324, 1997.

[Kohavi and Sahami, 1996] R. Kohavi and M. Sahami. Error-based and entropy-based discretization of continuous features. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 114–119. AAAI Press/MIT Press, 1996.

[Kohavi, 1995] R. Kohavi. The power of decision tables. In *Proceedings of the European Conference on Machine Learning*, pages 174–189. Springer Verlag, 1995.

[Kononenko *et al.*, 1984] I. Kononenko, I. Bratko, and E. Roskar. Experiments in automatic learning of medical diagnostic rules. Technical report, Ljubljana : Joseph Stefan Institute, Faculty of Electrical Engineering and Computer Science, 1984.

[Kurgan and Cios, 2004] L.A. Kurgan and .J. Cios. CAIM discretization algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 16(2) :145–153, 2004.

[Kwedlo and Kretowski, 1999] W. Kwedlo and M. Kretowski. An evolutionary algorithm using multivariate discretization for decision rule induction. In *Principles of Data Mining and Knowledge Discovery*, pages 392–397, 1999.

[Langley and Sage, 1994] P. Langley and S. Sage. Induction of selective Bayesian classifiers. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 399–406. Morgan Kaufmann, 1994.

[Langley *et al.*, 1992] P. Langley, W. Iba, and K. Thompson. An analysis of Bayesian classifiers. In *10th national conference on Artificial Intelligence*, pages 223–228. AAAI Press, 1992.

[Lechevallier, 1990] Y. Lechevallier. Recherche d'une partition optimale sous contrainte d'ordre total. Technical Report 1247, INRIA, 1990.

[LeCun *et al.*, 1995] Y. LeCun, L.D. Jackel, L. Bottou, A. Brunot, C. Cortes, J.S. Denker, H. Drucker, I. Guyon, U.A. Muller, E. Sackinger, P. Simard, and V. Vapnik. Comparison of learning algorithms for handwritten digit recognition. In F. Fogelman and P. Gallinari, editors, *International Conference on Artificial Neural Networks*, pages 53–60, 1995.

[Li and Vitanyi, 1997] M. Li and P.M.B. Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, Berlin, 1997.

[Liu *et al.*, 2002] H. Liu, F. Hussain, C.L. Tan, and M. Dash. Discretization : An enabling technique. *Data Mining and Knowledge Discovery*, 4(6) :393–423, 2002.

[Maass, 1994] W. Maass. Efficient agnostic pac-learning with simple hypothesis. In *COLT '94 : Proceedings of the seventh annual conference on Computational learning theory*, pages 67–75. ACM Press, 1994.

[Mamdouh, 2006] R. Mamdouh. *Data Preparation for Data Mining Using SAS*. Morgan Kaufmann Publishers, 2006.

[McCullagh, 1980] P. McCullagh. Regression model for ordinal data (with discussion). *Journal of the Royal Statistical Society B*, 42 :109–127, 1980.

[Meinshausen, 2006] N. Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7 :983–999, 2006.

[Monti and Cooper, 1999] S. Monti and G.F. Cooper. A latent variable model for multivariate discretization. In *The Seventh International Workshop on Artificial Intelligence and Statistics*, pages 249–254, 1999.

[Muhlenbach and Rakotomalala, 2002] F. Muhlenbach and R. Rakotomalala. Multivariate supervised discretization, a neighborhood graph approach. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, pages 314–321, 2002.

[Munteanu, 1996] P. Munteanu. *Extraction de connaissances dans les bases de données Parole : Apport de l'apprentissage symbolique*. PhD thesis, Institut National Polytechnique de Grenoble, 1996.

[Nadif and Govaert, 2005] M. Nadif and G. Govaert. Block clustering of contingency table and mixture model. In *Intelligent Data Analysis*, pages 249–259, 2005.

[Nagesh *et al.*, 2000] H. Nagesh, S. Goil, and A. Choudhary. A scalable parallel subspace clustering algorithm for massive data sets. In *International Conference on Parallel Processing*, pages 477–48, 2000.

[Neal, 1993] R.M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

[Pfahringer, 1995] B. Pfahringer. Compression-based discretization of continuous attributes. In *International Conference on Machine Learning*, pages 456–463, 1995.

[Provost *et al.*, 1998] F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 445–553, 1998.

[Pyle, 1999] D. Pyle. *Data preparation for data mining*. Morgan Kaufmann Publishers, Inc. San Francisco, USA, 1999.

[Quinlan, 1986] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1 :81–106, 1986.

[Quinlan, 1993] J.R. Quinlan. *C4.5 : Programs for Machine Learning*. Morgan Kaufmann, 1993.

[Raftery and Zheng, 2003] A.E. Raftery and Y. Zheng. Long-run performance of Bayesian model averaging. Technical Report 433, Department of Statistics, University of Washington, 2003.

[Reuters, 2000] Reuters. Reuters corpus, 2000. http ://about.reuters.com/-researchandstandards/corpus/.

[Rissanen, 1978] J. Rissanen. Modeling by shortest data description. *Automatica*, 14 :465–471, 1978.

[Ritschard *et al.*, 2001] G. Ritschard, D. A. Zighed, and N. Nicoloyannis. Maximisation de l'association par regroupement de lignes ou de colonnes d'un tableau croisé. *Mathématiques et Sciences Humaines*, 154-155 :81–98, 2001.

[Ritschard, 2002] G. Ritschard. Performance d'une heuristique d'agrégation optimale bi-dimensionnelle. In *Extraction et gestion des connaissances (EGC 2002)*, pages 185–196, 2002.

[Ritschard, 2003] G. Ritschard. Partition bic optimale de l'espace des prédicteurs. *Revue des Nouvelles Technologies de l'Information*, 1 :99–110, 2003.

[Robert, 2006] C.P. Robert. *Le choix bayésien Principes et pratique*. Springer, 2006.

[Schwartz, 1978] G. Schwartz. Estimating the dimension of a model. *Annals of statistics*, 6(2) :461–464, 1978.

[Scott, 1979] D.W. Scott. Averaged shifted histograms : Effective nonparametric density estimators in several dimensions. *Annals of statistics*, 13(3) :1024–1040, 1979.

[Shannon, 1948] C.E. Shannon. A mathematical theory of communication. Technical report, Bell systems technical journal, 1948.

[Shashua and Levin, 2002] A. Shashua and A. Levin. Ranking with large margin principles : two approaches. In *Proceedings of the Fiveteenth Annual Conference on Neural Information Processing Systems (NIPS)*, 2002.

[Slonim and Tishby, 2000] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In *Research and Development in Information Retrieval*, pages 208–215, 2000.

[Sturges, 1926] H.A. Sturges. The choice of a class interval. *Journal of the American Statistical Association*, 21 :65–66, 1926.

[Takeuchi *et al.*, 2006] I. Takeuchi, Q.V. Le, T.D. Sears, A.J., and Smola. Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7 :1231–1264, 2006.

[Vapnik, 1996] V. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, New-York, 1996.

[Vitányi and Li, 2000] P.M.B. Vitányi and M. Li. Minimum description length induction, bayesianism, and Kolmogorov complexity. *IEEE Transactions on information theory*, 46 :446–464, 2000.

[Witten and Frank, 2000] I.H. Witten and E. Frank. *Data Mining*. Morgan Kaufmann, 2000.

[Yang and Webb, 2002] Y. Yang and G. Webb. A comparative study of discretization methods for naive-Bayes classifiers. In *Proceedings of the Pacific Rim Knowledge Acquisition Workshop*, pages 159–173, 2002.

[Zighed and Rakotomalala, 1996] D.A. Zighed and R. Rakotomalala. Sipina-w(c) for windows : User's guide, 1996. Laboratory ERIC, University of Lyon 2.

[Zighed and Rakotomalala, 2000] D.A. Zighed and R. Rakotomalala. *Graphes d'induction*. Hermes, France, 2000.

[Zighed *et al.*, 1998] D.A. Zighed, S. Rabaseda, and R. Rakotomalala. Fusinter : a method for discretization of continuous attributes for supervised learning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(33) :307–326, 1998.

[Zighed *et al.*, 2005] D.A. Zighed, G. Ritschard, W. Erray, and V.M. Scuturici. Decision trees with optimal joint partitioning. *International Journal of Intelligent System*, 20(7) :693–718, 2005.

Titre. Recherche d'une représentation des données efficace pour la fouille des grandes bases de données

Résumé. La phase de préparation du processus de fouille des données est critique pour la qualité des résultats et consomme typiquement de l'ordre de 80% d'une étude. Dans cette thèse, nous nous intéressons à l'évaluation automatique d'une représentation, en vue de l'automatisation de la préparation des données.

A cette fin, nous introduisons une famille de modèles non paramétriques pour l'estimation de densité, baptisés modèles en grille. Chaque variable étant partitionnée en intervalles ou groupes de valeurs selon sa nature numérique ou catégorielle, l'espace complet des données est partitionné en une grille de cellules résultant du produit cartésien de ces partitions univariées. On recherche alors un modèle où l'estimation de densité est constante sur chaque cellule de la grille.

Du fait de leur très grande expressivité, les modèles en grille sont difficiles à régulariser et à optimiser. Nous avons exploité une technique de sélection de modèles selon une approche Bayesienne et abouti à une évaluation analytique de la probabilité a posteriori des modèles. Nous avons introduit des algorithmes d'optimisation combinatoire exploitant les propriétés de notre critère d'évaluation et la faible densité des données en grandes dimensions. Ces algorithmes ont une complexité algorithmique garantie, super-linéaire en nombre d'individus.

Nous avons évalué les modèles en grilles dans de nombreux contexte de l'analyse de données, pour la classification supervisée, la régression, le clustering ou le coclustering. Les résultats démontrent la validité de l'approche, qui permet automatiquement et efficacement de détecter des informations fines et fiables utiles en préparation des données.

Mots clés. Apprentissage, Exploration de données, Statistique Bayesienne, Préparation des données, Sélection de modèles.

Title. Search for an efficient data representation for mining large databases

Abstract. The data preparation step of the data mining process represents 80% of the problem and is both time consuming and critical for the quality of the modeling. In this thesis, our purpose is to design an evaluation criterion of data representations, in order to automate data preparation.

To overcome this problem, we introduce a non parametric family of density estimation models, named data grid models. Each variable is partitioned in intervals or in groups of values according to whether it is numerical of categorical, and the whole data space is partitioned into a grid of cells resulting from the cross-product of the univariate partitions. We then consider density estimation models where the density is assumed constant per data grid cell.

Because of their high expressiveness, data grid models are hard to regularize and to optimize. We exploit a model selection technique based on a Bayesian approach and obtain an exact analytic criterion for the posterior probability of data grid models. We introduce combinatorial optimization algorithms which leverage the properties of our evaluation criterion and the sparseness of data in large dimension. These algorithms have a guaranteed algorithmic complexity, which is super-linear with the sample size.

We evaluate data grid models in numerous tasks of data analysis, for supervised classification, regression, clustering or coclustering. The results demonstrate the validity of the approach, that allows to automatically and efficiently detect fine-grained and reliable information useful for the data preparation step.

Key words. Machine learning, Data exploration, Bayesianism, Data preparation, Model selection.