Clustering prédictif Décrire et prédire simultanément

Oumaima Alaoui Ismaili

NNT : 2016SACLA010

THÈSE DE DOCTORAT

DE

L'Université Paris–Saclay

PRÉPARÉE À

AgroParisTech

École doctorale 581 : Agriculture Alimentation BIologie Environnement Santé

Spécialité doctorale : *"Informatique appliquée"*

PAR

Mlle. Oumaima ALAOUI ISMAILI

Clustering prédictif
Décrire et Prédire simultanément

présentée et soutenue publiquement à Paris, le 10/11/16

Composition du Jury

Mme Christel VRAIN,	PR, LIFO - Université de Orléans	Rapportrice
M. Gilbert SAPORTA,	PR émérite, CNAM	Rapporteur
Mme. Chantal REYNAUD,	PR, Université Paris-Saclay	Examinatrice
M. Younès BENNANI,	PR, Université Paris 13	Président du Jury
M. Gilles BISSON,	Chargé de recherche, Université de Grenoble	Examinateur
M. Vincent LEMAIRE,	Ingénieur de recherche, Orange Labs Lannion	Co-directeur de thèse
M. Antoine CORNUÉJOLS,	PR, AgroParisTech – Université Paris-Saclay	Directeur de thèse

UMR MIA-Paris, AgroParisTech, INRA, Université Paris-Saclay
Orange Labs, équipe PROF, Lannion

NNT : 2016SACLA010

DOCTORAL THESIS

A THESIS SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY DELIVERED BY

L'Université Paris–Saclay

PREPARED AT

AgroParisTech

Doctoral school 581 : ABIES

SPECIALTY : *"Applied Computer Science"*

***Author*:**

Mlle. Oumaima ALAOUI ISMAILI

Predictive clustering
Describe and predict simultaneously

publicly defended in Paris, on 10/11/16

Composition of Jury

Mme Christel VRAIN,	PR, LIFO - Université de Orléans	Rapportrice
M. Gilbert SAPORTA,	PR émérite, CNAM	Rapporteur
Mme. Chantal REYNAUD,	PR, Université Paris-Saclay	Examinatrice
M. Younès BENNANI,	PR, Université Paris 13	Président du Jury
M. Gilles BISSON,	Chargé de recherche, Université de Grenoble	Examinateur
M. Vincent LEMAIRE,	Ingénieur de recherche, Orange Labs Lannion	Co-directeur de thèse
M. Antoine CORNUÉJOLS,	PR, AgroParisTech – Université Paris-Saclay	Directeur de thèse

UMR MIA-Paris, AgroParisTech, INRA, Université Paris-Saclay
Orange Labs, équipe PROF, Lannion

université PARIS-SACLAY

ÉCOLE DOCTORALE
Agriculture, alimentation, biologie, environnement, santé (ABIES)

Titre : Clustering prédictif : Décrire et Prédire simultanément

Mots clés : Description, prédiction, interprétation, clustering, classification supervisée, clustering prédictif

Résumé : Le clustering prédictif est un nouvel aspect d'apprentissage supervisé dérivé du clustering standard. Les algorithmes appartenant à ce type de l'apprentissage cherchent à décrire et à prédire d'une manière simultanée. Il s'agit de découvrir la structure interne d'une variable cible. Puis munis de cette structure, de prédire la classe des nouvelles instances.
Pour atteindre l'objectif de la thèse qui est la recherche d'un modèle d'apprentissage "interprétable" capable de décrire et de prédire d'une manière simultanée, nous avons choisi de modifier l'algorithme des K-moyennes standard. Cette version modifiée est nommée *les K-moyennes prédictives*. Ellle contient 7 différentes étapes dont chacune peut être supervisée indépendamment des autres.
Au cours de cette thèse, nous nous intéressons à la supervision de quatre étapes, à savoir : 1) le prétraitement des données, 2) l'initialisation des centres, 3) le choix de la meilleure partition et 4) la mesure d'importance des variables.
Nos résultats expérimentaux montrent d'une part qu'avec la supervision de l'étape de prétraitement des données et de l'étape d'initialisation des centres, l'algorithme des K-moyennes prédictives parvient à avoir des performances très compétitives ou meilleures que celles obtenues par certains algorithmes de clustering prédictif.
D'autre part, ces résultats expérimentaux mettent l'accent sur la capacité de nos méthodes de prétraitement à aider l'algorithme des K-moyennes prédictives à fournir des résultats facilement interprétables par l'utilisateur.
Nous montrons enfin dans ce mémoire qu'avec l'aide du critère d'évaluation proposé dans cette thèse, l'algorithme des K-moyennes prédictives parvient à sélectionner la partition optimale qui réalise le bon compromis entre la description et la prédiction. Ceci permet à l'utilisateur de découvrir les différentes raisons qui peuvent mener à une même prédiction.

Title : Predictive clustering : Describe and Predict simultaneously

Keywords : Description, prediction, interpretation, clustering, supervised learning, predictif clustering

Abstract : Predictive clustering is a new supervised learning framework derived from traditional clustering. This new framework allows to describe and to predict simultaneously. Compared to a classical supervised learning, predictive clsutering algorithms seek to discover the internal structure of the target class in order to use it for predicting the class of new instances.
The purpose of this thesis is to look for an interpretable model of predictive clustering. To acheive this objective, we choose to modified traditional K-means algorithm. This new modified version is called *predictive K-means*. It contains 7 differents steps, each of which can be supervised seperatly from the others. In this thesis, we only deal four steps : 1) data preprocessing, 2) initialization of centers, 3) selecting of the best partition, and 4) importance of features.
Our experimental results show that the use of just two supervised steps (data preprocessing and initialization of centers), allow the K-means algorithm to acheive competitive performances with some others predictive clustering algorithms.
These results show also that our preprocessing methods can help predictive K-means algorithm to provide results easily comprehensible by users.
We are also showing in this thesis that the use of our new measure to evaluate predictive clustering quality, helps our predictive K-means algorithm to find the optimal partition that establishes the best trade-off between description and prediction. It thus allows users to find the different reasons behind the same prediction : two differents instances could have the same predicted label.

Sommaire

Chapitre 3
Distance dépendante de la classe

Chapitre 4
Initialisation des centres

Chapitre 1

Introduction

1.1 Cadre de la thèse

De nos jours, les données sont devenues l'un des atouts majeurs qui constituent la richesse des entreprises. Les informations présentes, mais noyées dans la grande masse de données, sont devenues pour ces entreprises un facteur de compétitivité et d'innovation. Par exemple, les grandes entreprises telles qu'Orange et Amazon peuvent avoir un aperçu sur les préférences de leurs clients à partir de leurs données de comportement. En général, ces données permettent à l'utilisateur de découvrir et d'expliquer certains phénomènes existants ou bien d'extrapoler des nouvelles connaissances à partir des informations présentes. Pour exploiter ces grandes masses de données, de nombreuses techniques d'apprentissage automatique ont été développées. Dans cette thèse, nous nous intéressons particulièrement aux techniques d'apprentissage supervisé et non supervisé qui ont historiquement permis d'organiser efficacement des ensembles de données de tailles importantes.

Dans le cadre de l'apprentissage non supervisé, le clustering dans cette thèse, un grand nombre d'algorithmes ont été proposés. Ils permettent de discerner des groupes d'instances homogènes et différent les un des autres : les instances de chaque cluster doivent être très similaires les unes des autres et très différentes des instances des autres clusters.

Dans le cadre de l'apprentissage supervisé, la classification supervisée dans cette thèse, plusieurs algorithmes ont été proposés. Ils permettent d'apprendre un modèle qui, à partir d'un ensemble de données étiquetées, prédit ultérieurement la classe des nouvelles instances pas encore étiquetées. Les algorithmes d'apprentissage supervisé peuvent être divisés en deux grandes familles. La première famille regroupe l'ensemble des modèles dont les résultats sont peu compréhensibles pour l'utilisateur, i.e. modèles boîtes noires. La deuxième famille d'algorithme, quant à elle, regroupe l'ensemble des modèles qui permettent de fournir des résultats interprétables à l'utilisateur, i.e. modèles transparents.

Depuis quelques années, un nouvel aspect de l'apprentissage supervisé a vu le jour. Ce type d'apprentissage englobe à la fois les caractéristiques de la classification supervisée et du clustering. Contrairement au clustering qui privilégie l'axe de description et à la classification supervisée qui privilégie l'axe de prédiction, ce type d'apprentissage cherche à *décrire et à prédire simultanément.*

Pour aboutir à cette fin, deux principales voies existent. La première voie désigne le passage de la prédiction vers la description (voie 1 de la figure 1.1). Il s'agit ici de modifier les algorithmes de la classification supervisée dans le but de leur permettre de bien décrire les données. La deuxième voie désigne le passage de la description vers la prédiction (voie 2 de la figure 1.1). Il s'agit de modifier les algorithmes du clustering afin d'en faire de bons classifieurs. Les algorithmes

d'apprentissage provenant de cette deuxième voie sont appelés les algorithmes du *clustering prédictif*.

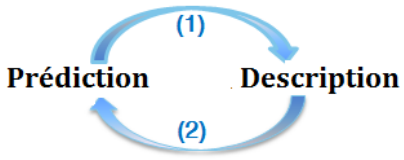


FIGURE 1.1 – Les deux voies principales pour décrire et prédire simultanément

Dans cette thèse, on s'intéresse exclusivement aux algorithmes du clustering prédictif. Notre objectif est donc la recherche d'un modèle de clustering prédictif "interprétable" capable de décrire et de prédire simultanément. Le modèle recherché est un modèle d'apprentissage qui permet de découvrir la structure interne de la variable cible. Puis, munie de cette structure de prédire la classe des nouvelles instances. Les résultats fournis par ce genre d'algorithme doivent impérativement être facilement compréhensibles par l'utilisateur.

L'algorithme modifié dans cette thèse est l'algorithme le plus populaire dans le domaine du clustering, à savoir l'algorithme des K-moyennes. Cette version modifiée est nommée au long de ce manuscrit *les K-moyennes prédictives*. Cette dernière contient 7 différentes étapes dont chacune peut être traitée et supervisée indépendamment des autres. Quatre de ces étapes de l'algorithme des K-moyennes ont été traitées dans cette thèse : 1) le prétraitement des données, 2) l'initialisation des centres, 3) le choix de la meilleure partition et 4) la mesure d'importance des variables après convergence.

1.2 Cas d'usage

Les algorithmes du clustering prédictif sont utilisés dans divers domaines d'application, notamment les domaines critiques (*e.g.,* service marketing Orange, domaine médical, les banques, *etc.*) où l'interprétation des résultats est une condition aussi importante que la performance du modèle. Ils permettent à l'utilisateur de connaître les différentes voies qui peuvent mener à une même prédiction.

À titre d'exemple, lorsque le service marketing Orange veut contacter des clients pour leur proposer un nouveau produit (*e.g.,* un forfait d'appel illimité à l'étranger), l'agent doit, avant ce contact commercial, construire des argumentaires commerciaux afin de les convaincre. Or, ces derniers sont fondés en se basant sur les motivations et les besoins des clients. Cependant, deux clients ayant une même prédiction (intéressés par le nouveau produit) peuvent avoir des motivations et des besoins différents : il se peut que le premier client soit un jeune étudiant ayant une famille à l'étranger et le deuxième client soit un homme d'affaire âgé ayant diverses collaborations à l'étranger. Bien que ces deux clients aient la même prédiction, les argumentaires commerciaux associés doivent certainement être différents. L'utilisation d'un l'algorithme de clustering prédictif à ce stade permettra à l'agent de connaître l'ensemble des motivations et des besoins des clients existant dans la base de données Orange. Cette information utile lui permettra par la suite de bien adapter ses argumentaires commerciaux vis-à-vis d'une cible clients.

1.3 Organisation du mémoire

La thèse se compose de cinq chapitres.

Le chapitre 2 est un état de l'art divisé en quatre parties correspondant aux thématiques principales de la thèse. La première partie est consacrée à l'axe de description. Une vision générale concernant les principales étapes du processus du clustering est introduite. La deuxième partie est consacrée à l'axe de prédiction. Après une brève présentation du principe de la classification supervisée, nous nous focalisons sur l'aspect interprétable de quelques modèles de la classification supervisée. La troisième partie, quant-à-elle, est consacrée à l'axe d'interprétation. Cette partie met l'accent sur les différentes directions d'interprétation ainsi que les différentes manières permettant une interprétation aisée des résultats issus des modèles d'apprentissage. Finalement, avant la présentation des objectifs de la thèse, la dernière partie présente une description non exhaustive des méthodes potentielles permettant de décrire et de prédire d'une manière simultanée.

Le chapitre 3 est consacré à l'étude de la première étape de l'algorithme des K-moyennes prédictives, à savoir, le prétraitement des données. L'intérêt de ce chapitre est de proposer une méthode de prétraitement capable d'aider l'algorithme des K-moyennes standard à atteindre l'objectif du clustering prédictif. L'algorithme des K-moyennes standard est basé sur une métrique qui permet de mesurer la proximité entre les instances indifféremment de leur classe d'appartenance : deux instances proches en termes de distance vont être considérées comme similaires bien qu'elles ont des étiquettes différentes. Ceci peut nuire à la performance du modèle au sens du clustering prédictif. L'idée présentée dans ce chapitre est l'utilisation d'une méthode de prétraitement capable d'écrire une distance dépendante de la classe. Cette méthode est capable de modifier indirectement la fonction du coût de l'algorithme des K-moyennes standard en lui permettant de ainsi d'établir une relation entre la proximité des instances en termes de distance et leur classe d'appartenance.

Le chapitre 4 est dédié à l'étude de la deuxième étape de l'algorithme des K-moyennes, à savoir, l'étape d'initialisation des centres. Le choix de la partition initiale a un impact direct sur la qualité des résultats issus de l'algorithme des K-moyennes. Dans le cadre du clustering prédictif, la qualité des résultats dépend de trois principaux facteurs : la compacité, la séparabilité et la pureté des clusters en termes de classes. Dans ce cadre, l'utilisation d'une méthode d'initialisation non supervisé va détériorer la qualité de l'algorithme des K-moyennes prédictives. En effet, dans le cas de déséquilibre des classes (*i.e.,* l'existence d'une classe majoritaire et d'une classe minoritaire), la probabilité de ne sélectionner aucun centre dans la classe minoritaire et de sélectionner plus qu'un centre dans la classe majoritaire est élevée. Par conséquent, une détérioration au niveau de la performance prédictive du modèle sera introduite. Dans ce contexte, nous proposons trois méthodes supervisées d'initialisation des centres dans le but d'aider l'algorithme des K-moyennes standard à surmonter ce problème.

Le chapitre 5 est dédié à l'étude de la quatrième étape de l'algorithme des K-moyennes, à savoir : le choix de la meilleure partition. L'algorithme des K-moyennes standard converge rarement vers un optimum global. Pour surmonter ce problème, cet algorithme doit être exécuté plusieurs fois dans le but de choisir la meilleure partition. Ce choix doit être effectué en utilisant un critère d'évaluation de la qualité. Selon notre connaissance, il n'existe dans la littérature un critère analytique permettant de mesurer le compromis entre la description et la prédiction. Dans ce cas, le choix de la meilleure partition devient une tâche difficile. Pour surmonter ce problème, nous proposons de modifier le critère d'évaluation Davies-Bouldin (DB) communément utilisé dans la littérature en lui intégrant une nouvelle mesure de dissimilarité "supervisée" capable

d'introduire une relation entre la proximité des instances en termes de distance et leur classe d'appartenance.

Le chapitre 6 présente d'une part une synthèse des résultats obtenus dans les chapitres précédents et d'autre part, il présente une comparaison de l'algorithme des K-moyennes prédictives proposé dans cette thèse (intégrant les différentes étapes supervisées) avec d'autres méthodes issus de la littérature. Cette partie expérimentale est divisée en deux grandes parties. La première partie se focalise sur le côté prédictif du modèle. Tandis que la deuxième partie se focalise sur l'aspect interprétable du modèle et la capacité de celui-ci à bien découvrir la structure interne de la variable cible. Pour finir, une conclusion dresse le bilan des trois années de thèse, des travaux réalisés et des travaux futurs. Nous rappelons les différentes notions introduites dans la thèse, ainsi que les résultats obtenus.

Un travail sur l'importance des variables a été aussi effectué dans cette thèse. Ce travail fait l'objet de deux publications. Néanmoins, comme il n'est pas totalement aboutit, le choix a été fait de ne pas l'incorporer au cœur du manuscrit de la thèse mais de le placer en Annexe E.

Chapitre 2

Entre la prédiction et la description

Sommaire

Ce chapitre a fait l'objet des publications suivantes :

[9] Oumaima Alaoui Ismaili, Vincent Lemaire, Antoine Cornuéjols. **Classification à base de clustering ou comment décrire et prédire simultanément ?**. Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJCIA), Rennes, pages 7-12, 2015.

[12] Oumaima Alaoui Ismaili, Vincent Lemaire, Antoine Cornuéjols : **Clustering prédictif : décrire, prédire et interpréter simultanément** à venir sur invitation suite à la conférence RJCIA, in Revue d'intelligence Artificielle (RIA).

2.1 Introduction

Au cours de ces dernières décennies, le monde a connu une véritable explosion du volume des données. La multiplication des systèmes et d'appareils capables de générer et de transmettre automatiquement des données est l'un des principaux facteurs à l'origine de ce phénomène. Chaque individu peut générer quotidiennement une multitude d'informations diverses et variées (*e.g.,* images, films, textes, sons, *etc.*) via le web, les réseaux sociaux et les appareils nomades. L'innovation continue des techniques de stockage figure également parmi les principaux facteurs de cette croissance exponentielle du volume des données. Par exemple, les grandes entreprises comme *Orange* et *Amazon* récoltent et stockent quotidiennement une avalanche de données concernant les comportements de leurs clients. Les résultats d'analyses médicales et les mesures effectuées un peu partout dans le monde comme les mesures météorologiques remplissent aussi d'importantes bases de données numériques.

Les données récoltées par ou pour les entreprises sont devenues un atout important. Les informations présentes, mais à découvrir au sein des grands volumes de données, sont devenues pour ces entreprises un facteur de compétitivité et d'innovation. Par exemple, à travers la connaissance des comportements des consommateurs, les entreprises peuvent avoir un aperçu de leurs attentes et de leurs besoins. L'étude des résultats médicaux peut également aider à mieux identifier les patients à risque, permettant ainsi de prévenir plutôt que de guérir. De ce fait, il existe un grand intérêt à développer des techniques permettant d'utiliser au mieux les gisements de données afin d'en extraire un maximum de connaissances utiles.

Dans la littérature, de nombreuses techniques d'analyse issues de diverses disciplines scientifiques (*e.g.,* statistique, Intelligence Artificielle, Informatique) ont été proposées. Par exemple, l'analyse multivariée [88] regroupe l'ensemble des méthodes statistiques qui s'attachent à l'observation et au traitement simultané de plusieurs variables en vue d'en dégager une information synthétique pertinente. Les deux grandes catégories de méthodes d'analyse statistique multivariées sont, d'une part, les méthodes dites *descriptives* et, d'autre part, les méthodes dites *prédictives*.

Les méthodes descriptives ont pour objectif d'organiser, de simplifier et d'aider à comprendre les phénomènes existant dans un ensemble important de données non étiquetées. Cet ensemble est organisé en instances constituées de plusieurs variables descriptives, où aucune des variables n'a d'importance particulière par rapport aux autres. Toutes les variables sont donc prises en compte au même niveau. Les trois grandes catégories de méthodes descriptives sont : *la description*, *la segmentation* et *l'association*.

1. *La description* [88] consiste à dégager les aspects les plus intéressants de la structure des données. Par exemple, les techniques d'analyse factorielles consistent à dégager des variables cachées dites "*facteurs*" à partir d'un ensemble de mesures. L'utilité de ces facteurs réside dans le fait qu'un nombre réduit de ces derniers explique aussi bien les données que l'ensemble des variables descriptives. Parmi les techniques factorielles, on citera celles les plus connues : Analyse en Composantes Principales (ACP) pour les variables quantitatives, Analyse des Correspondances Multiples (ACM) pour les variables qualitatives, Analyse Factorielle des Correspondances (AFC) pour les variables qualitatives et Analyse Factorielle Multiple (AFM) pour des groupes de variables quantitatives et/ou qualitatives.
2. *La segmentation* (le clustering ou la classification non supervisée) [2, 88, 62] cherche à discerner une structure dans un ensemble de données non étiquetées. L'objectif est de trouver une typologie ou une répartition des individus en groupes distincts. Chaque groupe (ou

cluster) doit contenir les individus les plus homogènes possible. Il s'agit donc de construire un modèle permettant de mieux présenter les observations de manière à la fois précise et compacte (voir section 2.2). Parmi les méthodes permettant d'atteindre cet objectif, on trouve par exemple : l'algorithme des K-moyennes, la classification hiérarchique ascendante/descendante et les réseaux de Kohonen, *etc.*

3. *L'association* consiste à mesurer le degré d'association entre deux ou plusieurs variables. Les relations découvertes sont exprimées sous forme de règles d'association. Cette analyse est appelée aussi analyse d'affinité. Elle est très utile par exemple pour détecter les produits achetés simultanément, dans une grande surface, par un très grand nombre de clients. Cette information sert à mieux fixer les assortiments et les offres promotionnelles. Les algorithmes utilisés dans ce cadre ont comme principe de détecter les propriétés qui reviennent fréquemment dans l'ensemble des données afin d'en déduire une catégorisation. Dans ce cadre d'étude, l'algorithme Apriori [3] est l'algorithme le plus utilisé.

Les méthodes prédictives permettent de prévoir et d'expliquer à partir d'un ensemble de données étiquetées un ou plusieurs phénomènes observables. Dans ce cadre, deux types de techniques se distinguent : *la régression* et *la classification supervisée.*

1. *La régression* a pour but de trouver à partir d'un ensemble de données, le lien entre les prédicteurs et une variable cible "numérique" à prédire. Parmi les méthodes permettant d'atteindre cet objectif, on trouve par exemple : la régression linéaire simple, la régression multiple, la régression logistique et le modèle linéaire généralisé (GLM) [88, 35], *etc.*
2. *La classification supervisée* est une estimation qui consiste à découvrir le lien entre une variable cible "catégorielle" et des variables descriptives. L'idée de base est de proposer un modèle permettant de prévoir l'appartenance des nouveaux individus à des classes prédéterminées. Les méthodes les plus répandues dans ce cadre sont : les réseaux de neurones (ANN), les machines à vecteurs de support (SVM) et forêts aléatoires (RF) [35, 68].

Dans la littérature sur le sujet d'extraction des connaissances utiles, le terme d'**apprentissage automatique** est souvent utilisé. Comme l'indique son nom, cette technique consiste à programmer la machine pour qu'elle apprenne à effectuer des tâches difficiles à travers des moyens algorithmiques. L'idée de base est de construire un modèle à partir d'un jeu de données, duquel les performances peuvent être évaluées en utilisant des méthodes de validation. Ces méthodes diffèrent selon le type d'apprentissage suivi (*e.g.,* la précision pour la classification supervisée et l'inertie intra\inter clusters pour le clustering). L'apprentissage automatique se décline en plusieurs variantes en fonction de la nature des données dont on dispose (supervisé, non supervisé, *etc.*). On peut donc placer la classification supervisée dans le domaine de l'apprentissage supervisé et le clustering dans le domaine de l'apprentissage non supervisé.

Dans cette thèse, nous nous intéressons exclusivement à la classification supervisée et non supervisée qui ont historiquement permis d'extrapoler de nouvelles informations à partir des informations présentes ou bien de découvrir et d'expliquer certains phénomènes existants mais noyés dans le volume de données.

Depuis quelques années, les chercheurs ont concentré leur attention sur l'étude d'un nouvel aspect d'apprentissage. Ce dernier fusionne à la fois les caractéristiques de la classification supervisée (la prédiction) et du clustering (la description). Les algorithmes appartenant à ce type d'apprentissage cherchent à **décrire et à prédire simultanément**. Il s'agit ici de découvrir la structure interne de la variable cible. Puis, munis de cette structure, de prédire la classe des nouvelles instances. Cette technique permet à l'utilisateur d'améliorer sa compréhension vis-à-vis des données. En effet, contrairement à la classification supervisée, les algorithmes descriptifs et prédictifs à la fois permettent à l'utilisateur de connaître les différentes voies qui peuvent mener à une même prédiction : deux instances très différentes peuvent avoir la même prédiction de classe. L'obtention d'une telle information est très utile dans plusieurs domaines d'application, notamment, dans les domaines critiques où l'interprétation des résultats issus des algorithmes d'apprentissage est une condition primordiale. A titre d'exemple, dans le domaine médical, deux patients X_1 et X_2 ayant comme prédiction un test positif (la classe $\{+\}$ de la figure 2.1) pour l'AVC (*i.e.,* une grande probabilité d'avoir un Accident Vasculaire Cérébral) n'ont pas forcément les mêmes causes et\ou les mêmes symptômes de l'AVC : il se peut que le patient X_1 soit une personne âgée, qui souffrait de la fibrillation auriculaire et qui par conséquent a eu des maux de têtes et des difficultés à apprendre (par exemple, X_1 appartient au groupe A de la figure 2.1). Tandis que le patient X_2, pourrait être une jeune personne qui consommait de l'alcool d'une manière excessive et, par conséquent a perdu l'équilibre (par exemple, X_2 appartient au groupe B de la figure 2.1).

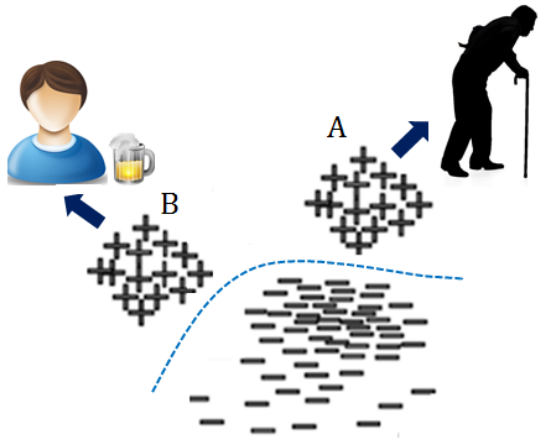


FIGURE 2.1 – Problème d'une classification binaire

L'objectif de cette thèse est la recherche d'un algorithme d'apprentissage "interprétable" permettant de décrire et de prédire d'une manière simultanée. Il s'agit ici de trouver un modèle capable d'équilibrer les trois axes (description, prédiction et interprétation) comme le montre la figure 2.2.

Pour atteindre cet objectif, il existe deux voies principales, à savoir : 1) rendre les méthodes descriptives plus prédictives ou 2) rendre les méthodes prédictives plus descriptives. Ceci est effectué en respectant l'axe d'interprétation. Avant d'entamer cette problématique dans la section 2.5, il est intéressant tout d'abord d'avoir une vision globale sur la classification supervisée, le clustering et l'interprétation des résultats issus d'un modèle d'apprentissage.

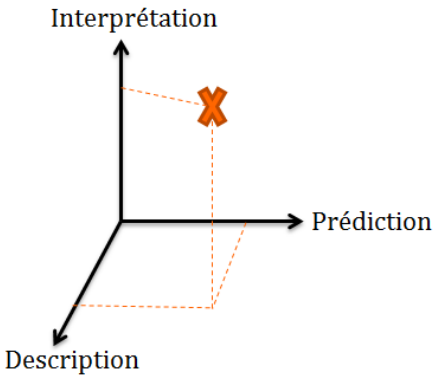


FIGURE 2.2 – Les trois axes traités dans cette thèse

Le reste de ce chapitre est donc organisé comme suit : La section 2.2 présente les trois étapes principales qui forment le processus du clustering. La section 2.3 met l'accent sur la capacité des modèles de classification supervisée à générer des résultats facilement interprétable par l'utilisateur. Cette section est divisée en deux parties principales. La première partie est dédiée aux modèles naturellement interprétables (voir Section 2.3.1). La deuxième partie (voir Section 2.3.2), quant à elle, présente l'ensemble des techniques qui peuvent aider à interpréter facilement les résultats générés par les modèles boîtes noires (i.e., les modèles fournissant des résultats incompréhensibles par l'être humain). Puisque l'interprétation est une notion clé dans cette thèse, la section 2.4 présentera d'une manière générale ses différents aspects. La section 2.5, quant à elle, met l'accent sur un nouveau type d'apprentissage qui consiste à fusionner les caractéristiques de la classification supervisée et du clustering. Finalement, la section 2.6 se focalise sur la présentation de notre problématique, de nos objectifs et des propositions préliminaires.

2.2 Approche descriptive : *La classification non supervisée*

Les approches descriptives désignent l'ensemble des méthodes permettant d'organiser et d'identifier des tendances dans les données. Loin de la volonté de faire un état de l'art exhaustif de toutes les méthodes descriptives existantes, on s'intéresse dans cette section uniquement à l'une des moyennes utilisée pour décrire les données, à savoir, *le clustering*. Cette section en présente ses concepts clefs.

Le clustering consiste à trouver la distribution sous-jacente des exemples dans leur espace de description. Autrement dit, à partir d'une base de données non étiquetées, cette approche vise à former des groupes (ou clusters) homogènes en fonction d'une certaine notion de similarité. Les observations qui sont considérées similaires sont associées au même groupe alors celles qui sont considérées comme différentes sont associées à des groupes différents. Plus formellement, dans les problèmes de clustering, les données $\mathcal{D} = \{X_i\}_{i=1}^{N}$ sont composées de N observations sans étiquette (ou classe), chacune décrite par plusieurs variables. On notera $X_i = \{X_i^1, \ldots, X_i^d\}$, l'ensemble de d variables décrivant l'observation i ($i \in [1, N]$). L'objectif ici est donc de partitionner l'espace d'entré en K clusters. Chaque cluster S_k ($k \in \{1, \ldots, K\}$) doit être, d'une part, différent des autres clusters et d'autre part, doit contenir des observations similaires.

D'une manière générale, le processus du clustering se divise en trois étapes principales (voir Figure 2.3) : (1) La préparation des données, (2) Le choix de l'algorithme de clustering et (3) La validation et l'interprétation des résultats.

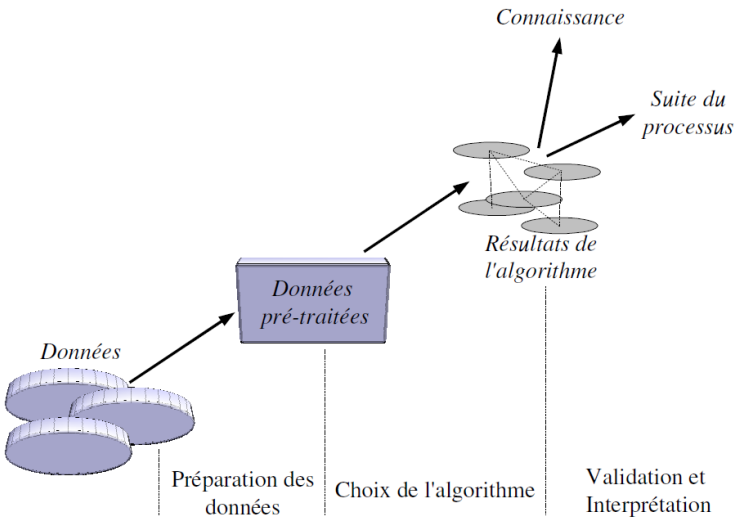


FIGURE 2.3 – Les différentes étapes du processus de clustering

2.2.1 La préparation des données

La phase de préparation des données est particulièrement importante dans le processus de la classification non supervisée (*e.g.,* [77] [30]). Bien souvent, une mauvaise représentation produit un clustering complexe et difficilement exploitable. Les données initiales peuvent contenir du bruit, des outliers, provenir de variables natives ou construites, *etc.* L'objectif du prétraitement est donc de chercher une représentation des données performante dans le contexte d'étude. La phase de préparation des données comprend des étapes de sélection, nettoyage, construction, intégration et recodage des données. Généralement, un bon prétraitement peut permettre au modèle d'identifier des clusters intéressants. Cependant, il se peut que celui-ci empêche l'interprétation ultérieure des résultats. Un état de l'art bien détaillé de ce sujet sera présenté dans le chapitre 2 de ce mémoire.

2.2.2 Le choix de l'algorithme de clustering

Le choix de l'algorithme de clustering dépend en général de la nature des variables (*e.g.,* quantitative et qualitative) dans les données et des clusters attendus (*e.g.,* nombre, forme, densité, *etc*). Généralement, les critères de décision peuvent être :

- *Les connaissances a priori* définissent l'ensemble d'informations concernant le nombre de clusters désiré, la distance minimale entre les clusters disjoints, *etc.*
- *La présentation des résultats* définit le type de la sortie de l'algorithme (*e.g.,* une hiérarchie de clusters ou une partition de l'ensemble des exemples).
- *La complexité* représente le temps de calcul nécessaire à la résolution d'un problème. Il est un critère important à prendre en compte lors du choix de l'algorithme. En particulier, il est admis que la complexité algorithme doit être linéaire en fonction du nombre d'exemples dans le cas des bases de données volumineuses.
- *Déterministe* définit la capacité des algorithmes à fournir les mêmes résultats (sans aucun changement) en utilisant les mêmes données en entrée.
- *Incrémental* définit la manière dont les données sont intégrées dans l'algorithme. Dans ce cas, les données sont intégrées au fur et à mesure de leur arrivée.
- *Prise en compte du contexte* définit la capacité de l'algorithme à prendre en compte ou

non la problématique du contexte.
— *La tolérance au bruit* définit la capacité de l'algorithme à gérer ou non le bruit qui peut exister dans les données.
— *La tolérance aux clusters de tailles variées* définit la capacité de l'algorithme à détecter des clusters ayant des tailles différentes (Figure 2.4).
— *La tolérance aux clusters de densités variées* définit la capacité de l'algorithme à réaliser des clusters ayant des densités différentes (Figure 2.4).

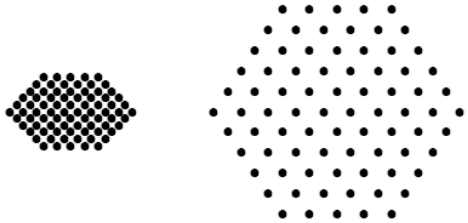


FIGURE 2.4 – Clusters de tailles et de densités différentes

— *La tolérance aux clusters de formes quelconques* définit la capacité de l'algorithme à réaliser des clusters ayant des formes différentes (Figure 2.5).
— *La tolérance aux clusters concentriques* définit la capacité de l'algorithme à réaliser des clusters concentriques, c'est-à-dire, inscrits les uns dans les autres (Figure 2.5).

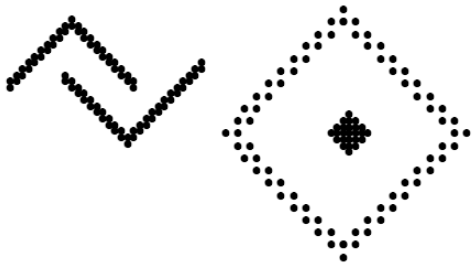


FIGURE 2.5 – Clusters de formes variées et concentriques

Le clustering se catégorise en plusieurs familles de méthodes selon la stratégie suivie pour construire les clusters. Parmi ces méthodes on trouve :

• **Le clustering hiérarchique**, comme l'indique son nom, cette approche consiste à former une hiérarchie de clusters : plus on descend dans la hiérarchie, plus les groupes sont spécifiques à un certain nombre d'exemples considérés comme similaires. Le clustering hiérarchique se catégorise en deux grandes familles : les méthodes '*ascendantes*' et les méthodes '*descendantes*'.

1. *Les méthodes ascendantes* commencent par une solution spécifique aux données pour arriver à une autre plus générale. Les méthodes de cette catégorie démarrent avec autant d'exemples que de clusters. Ensuite, elles fusionnent à chaque étape des clusters selon un critère donné jusqu'à l'obtention d'un seul cluster contenant ainsi l'ensemble de données.
2. *Les méthodes descendantes* partent d'une solution générale vers une autre plus spécifique. Les méthodes de cette catégorie démarrent avec un seul cluster contenant la totalité

des données, ensuite, elles divisent à chaque étape les clusters selon un critère jusqu'à l'obtention d'un ensemble de clusters différents stockés aux feuilles de la hiérarchie.

Il existe différentes approches pour mesurer la distance entre les clusters. On citera à titre d'exemple :

— L'approche *single-link* définit la distance entre deux clusters comme étant le minimum des distances pour toutes les paires d'exemples appartenant à des clusters différents [2].
— L'approche *complete-link* définit la distance entre deux clusters comme étant le maximum des distances pour toutes les paires d'exemples appartenant à des clusters différents [2].
— L'approche *average-link* définit la distance entre deux clusters comme étant la moyenne des distances pour toutes les paires d'exemples appartenant à des clusters différents [2].

• **Le clustering par partitionnement** consiste à diviser de manière optimale l'ensemble des instances en un groupe fini de groupes (K). L'objectif est ici de minimiser une mesure de la dissemblance intra-groupe pour k groupes. Le problème étant lié à l'optimisation d'une combinatoire, la solution trouvée sera rarement l'optimum global mais plutôt un des nombreux optimums locaux. Parmi ces méthodes, on trouve : les K-moyennes, les K-médoïdes ou le partitionnement autour des médoïdes (PAM) et la Carte Auto-Organisatrice.

• **Le clustering spectral** est considéré également comme un clustering de partitionnement. Par rapport à des algorithmes classiques comme celui des K-moyennes, cette technique offre l'avantage de classer des ensembles de données de structure « non-globulaire » dans un espace de représentation adéquat.

• **Le clustering basé sur la densité** consiste à identifier dans l'espace de description des objets les régions de forte densité, entourées par des régions de faible densité pour former les clusters.

• **Le clustering basé sur les grilles** consiste à partitionner l'espace en différentes cellules à l'aide d'une grille, puis à identifier les ensembles de cellules denses connectées pour la formation des clusters. Les méthodes appartenant à cette catégorie nécessitent deux paramètres à savoir : la taille de la grille et la densité minimum déterminant si une cellule de la grille est considérée comme dense ou non.

• **Le clustering basé sur les graphes** considère les clusters comme étant des ensembles de nœuds connectés dans un graphe. L'objectif est donc de former le graphe qui connecte les ensembles entre eux de telle manière que la somme des valeurs des arcs correspondant aux distances entre les exemples soit minimale.

Des états de l'art plus détaillés sont disponibles dans la littérature. Le lecteur souhaitant une description plus avancée des méthodes pourra s'y référer ([2],[88], [62]). Le tableau 2.1 présente à titre illustratif un ensemble de caractéristiques associées à certaines méthodes de clustering. Il est à noter qu'aucune méthode de clustering n'est intrinsèquement meilleure que les autres sur l'ensemble des problèmes envisageables.

Caractéristiques	**Hiérarchique**	**K-moyennes**
Connaissances a priori	Nombre de clusters K	Nombre de clusters K
Présentation des résultats	Hiérarchie	K centroïdes
Complexité	$O(M \times N^2)$	$O(M \times N \times K)$
Déterministe	oui	non
Incrémental	non	oui
Prise en compte du contexte	non	non
Tolérance au bruit	non	non
Tolérance aux clusters de tailles variées	oui	non
Tolérance aux clusters de densités variées	oui	oui
Tolérance aux clusters de forme quelconque	oui	non
Tolérance aux clusters concentriques	oui	non

Caractéristiques	**Basé sur la densité**	**Basé sur la grille**
Connaissances a priori	Critère de densité du voisinage	Taille de grille et critère de densité
Présentation des résultats	Partition	Ensemble de cellules connectées
Complexité	$O(M \times N^2)$	$O(M\times$ taille de grille$)$
Déterministe	oui	oui
Incrémental	non	non
Prise en compte du contexte	non	non
Tolérance au bruit	oui	oui
Tolérance aux clusters de tailles variées	oui	oui
Tolérance aux clusters de densités variées	non	non
Tolérance aux clusters de forme quelconque	oui	oui
Tolérance aux clusters concentriques	oui	oui

TABLE 2.1 – Caractéristiques associées aux différentes méthodes de clustering.
N = Le nombre d'observations.
d = Le nombre de variables.
K = Le nombre de clusters.

2.2.3 Validation et Interprétation des résultats

L'évaluation de la pertinence de la partition générée par le clustering est un domaine de recherche très actif. La difficulté de ce problème réside dans le fait que l'évaluation des résultats du clustering est en partie subjective [53]. En effet, pour un même jeu de données, il existe souvent un grand nombre de partitions possibles. De plus, il est impossible de définir un critère universel permettant d'évaluer sans biais les résultats obtenus par l'ensemble des algorithmes de clustering. Pour atteindre cet objectif, de nombreuses techniques ont été développées pour identifier la « meilleure » partition générée par un algorithme de clustering. Cette identification est souvent liée à la méthode utilisée.

Dans la littérature, les critères analytiques permettant de mesurer la qualité des résultats issus des algorithmes de clustering peuvent être catégorisés en trois grandes familles : *interne, externe* et *relatif.*

1. *Les mesures de qualité interne* [71] se calculent uniquement à partir des informations contenues dans les données, sans avoir recours à des connaissances a priori. Ces mesures sont en général des approches non supervisées qui se basent sur des informations internes au clustering. Ces mesures se basent souvent sur la définition intuitive et la plus simple du clustering : les groupes d'instances doivent être les plus compacts possibles (*i.e.,* la similarité intra clusters) et différents les uns des autres (*i.e.,* la similarité inter clusters).

Les mesures internes permettent donc d'évaluer *la compacité* et *la séparabilité* des clusters. Les mesures de qualité internes les plus connues dans la littérature sont : l'indice *Davies Bouldin* [38], *indice SD* [55], *indice Silhouette* [91] *etc.*

2. *Les mesures de qualité externe* [45] s'appuient sur une connaissance a priori des caractéristiques d'un bon clustering. Ces mesures sont en général des approches supervisées qui consistent à mesurer le degré de correspondance entre la partition générée par l'algorithme de clustering et une partition connue des données. De nombreuses mesures ont été proposées pour atteindre cet objectif. Citons par exemple, l'indice Adjusted Rand Index (ARI) [57], l'information mutuelle normalisée (NMI) [96] et l' entropie conditionnelle [45].
3. *Les mesures de qualité relative* permettent de comparer plusieurs partitionnements obtenus à partir d'un même jeu de données. Il s'agit tout simplement de l'utilisation des critères internes ou externes pour choisir la meilleure partition générée par le même algorithme sur le même jeu de données.

Selon les besoins de l'utilisateur, le clustering peut être utilisé pour deux différentes raisons :

1. La tâche de clustering peut être inscrite comme une étape intermédiaire dans un traitement d'apprentissage (voir Figure2.6) : il s'agit ici de considérer le clustering comme une étape de prétraitement utilisée dans une autre tâche telle que la classification supervisée. Une description des clusters (*i.e.,* l'interprétation des résultats générés par le clustering) n'est pas nécessaire dans cette situation. On cherche uniquement dans ce cas à obtenir l'appartenance des observations à l'un des clusters (ID-cluster) sans avoir besoin d'interpréter les résultats issus de l'algorithme. Le point le plus important ici est de s'assurer que la qualité des résultats fournis par l'algorithme est bonne. Pour se faire, les critères d'évaluation cités ci-dessus peuvent être utilisés (*e.g.,* Davies-Boulin "DB", Silhouette, etc).

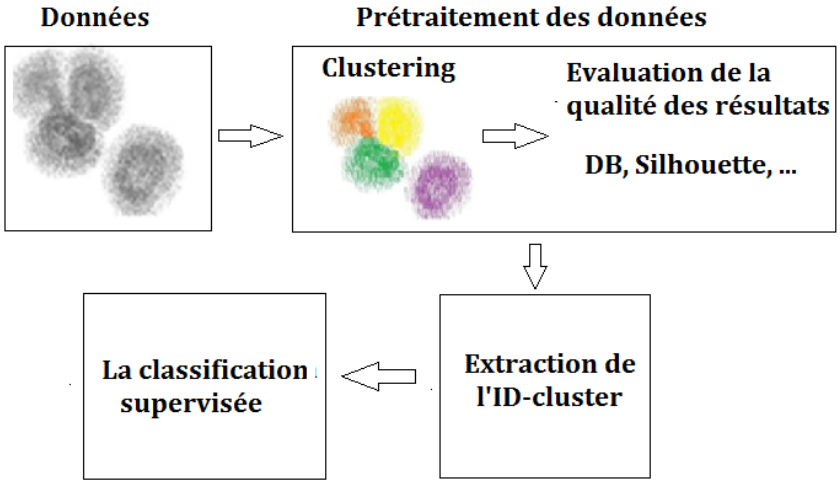


FIGURE 2.6 – Le clustering est considéré comme une étape de prétraitement.

2. Les clusters générés par le clustering constituent un résultat final (Voir Figure2.7). Dans ce cas, le clustering constitue à lui seul un processus global de découverte de groupes. L'exploitation des clusters pour une application donnée passe alors par une description de ces derniers.

FIGURE 2.7 – Le clustering constitue un résultat final

2.3 Approche prédictive : *La classification supervisée*

La classification supervisée cherche à prédire la classe des nouvelles instances en se basant sur des informations connues *a priori*. Elle est un processus à deux étapes : une étape d'apprentissage et une étape de classification.

Dans l'étape d'apprentissage, un modèle est construit en analysant un jeu de données dit *"d'apprentissage"* dans lequel la classe de chaque instance est supposée prédéfinie. Soit $\mathcal{D} = \{(X_i, Y_i), i \in \{1, ..., N\}\}$ un jeu de données d'apprentissage composé de N instances. Chaque instance $(X_i = \{X_i^1, ..., X_i^d\}, Y_i \in \{1, ..., J\})$ est représentée par un vecteur de variables de dimension d et d'une variable cible Y_i indiquant son appartenance à une des J classes. Soit χ et κ respectivement l'espaces des valeurs d'entré et de sortie. D'une manière plus formelle, l'étape d'apprentissage a pour but d'apprendre, à partir des données d'apprentissage, une fonction $f : \chi \rightarrow \kappa$ de telle sorte que $f(X)$ est un "bon" prédicteur de la valeur correspondante à Y.

Dans l'étape de classification, le modèle construit dans la première étape est utilisé pour classer les nouvelles instances.

Le modèle construit par un algorithme d'apprentissage doit en général remplir un certain nombre de critères. Citons à titre d'exemple :

- Le taux d'erreur doit être le plus bas possible. Ce point peut être mesuré en utilisant plusieurs critères d'évaluation. A titre d'exemple, la précision (ACC), l'air sous la courbe de ROC (AUC), l'indice ARI (Adjusted Rand Index),..., *etc.*
- Il doit être aussi peu sensible que possible aux fluctuations aléatoires des données d'apprentissage.
- les décisions de classification doivent autant que possible être explicites et compréhensibles.

Le tableau 2.2 présente une comparaison de quelques modèles de la classification supervisée en se basant sur quelques critères de pertinence. Des états de l'art plus détaillés sont disponibles, le lecteur souhaitant une description plus avancée de ces modèles pourra s'y référer ([35, 68]).

Loin de vouloir donner une description détaillée des différentes méthodes de la classification supervisée, cette section traite exclusivement l'aspect interprétable de quelques méthodes. Dans

Critère	Arbre de décision	SVM	Plus proche voisin	Bayésien naïf
Rapidité d'apprentissage	+	- -	+ +	+
Rapidité et facilité de mise à jour	- -	-	+ +	+ +
Précision	+ +	+ +	+	+
Simplicité (nombre de paramètres)	-	-	+ +	+ +
Rapidité de classement	+ +	-	-	+ +
Interprétabilité	++	-	+ +	+ +
Généralisation - Sensibilité au bruit	-	+	-	+ +

TABLE 2.2 – Comparaison de quelques méthodes de classification suivant quelques critères de pertinence

la littérature, les modèles de la classification supervisée se catégorisent en deux grandes familles : les modèles transparents et les modèles boîtes noires. Les modèles transparents désignent tous les algorithmes d'apprentissage qui fournissent des résultats facilement interprétables par l'utilisateur. Contrairement aux modèles boîtes noires (ou opaque) qui désignent les algorithmes d'apprentissage fournissant des résultats non difficilement compréhensibles par l'utilisateur.

2.3.1 Les modèles transparents

Dans cette section, nous présentons les facteurs principaux permettant aux modèles tels les arbres de décision de générer des résultats compréhensibles par l'utilisateur.

1. **Les arbres de décision** ([87],[25]) figurent parmi les modèles naturellement interprétables. Ils fournissent des règles faciles à interpréter par l'être humain. En effet, l'arbre de décision est en général un classifieur présenté sous forme d'une structure arborescente (*e.g.,* voir la figure 2.8). Chaque nœud de l'arbre est : *i*) soit un nœud "de décision" où des tests ont été effectués sur les valeurs d'une seule variable. *ii*) Soit un nœud "feuille" qui détient la prédiction de la classe. Par conséquent, des règles inductives sont crées pour tous les chemins possibles de la racine à une des feuilles.

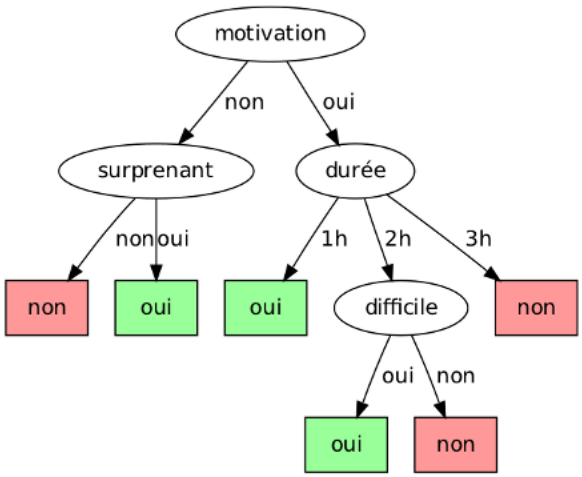


FIGURE 2.8 – Exemple d'arbre de décision pour la question "la présentation est-elle intéressante ?"

2. **Les K plus proches voisins** [37] est un classifieur qui ne nécessite pas d'apprendre une fonction précise d'apprentissage pour qu'il prédise la classe des nouvelles instances. Dans le cas des jeux de données de petites dimensions, ce modèle a la capacité de fournir à l'utilisateur un certain type explication concernant la classification de chaque nouvelle instance. Ces explications sont obtenues à l'aide d'une analyse simple des K plus proches voisins utilisés pour classer une instance. Il est à signaler que dans le cas des jeux de données de grandes dimensions, l'algorithme des K-plus proches voisins devient un modèle boîte noire.

Pour plus de détails sur les différentes raisons permettant à ces modèles d'être interprétables, le lecteur peut se référer à l'article [50].

2.3.2 Interprétation des modèles boîtes noires

Dans certains cas d'étude, l'interprétation des résultats d'un classifieur reste une question secondaire. La performance prédictive du modèle est dans cette situation le point clé pour la résolution des problématiques. Les algorithmes utilisés dans ce cadre privilégient plus le critère de performance prédictive que celui de l'interprétation. Ces modèles sont connus sous le nom *des modèles boîtes noires*. Parmi ces méthodes, on pourra citer notamment les réseaux de neurones (ANN) et les machines à vecteurs de support (SVM).

Les résultats fournis par les modèles boîtes noires sont incompréhensibles et ne conduisent donc pas à des interprétations informatives. Seule une étude des erreurs de prévisions permet de se faire une idée de la qualité du modèle en question. À titre d'exemple :

— *Les réseaux de neurones* (ANN) reçoivent les informations sur une couche réceptrice de "neurones". Ils traitent tout d'abord ces informations avec ou sans l'aide d'une ou plusieurs couches "cachées" contenant un ou plusieurs neurones. Ensuite, ils produisent un (ou plusieurs) signaux de sortie. Généralement, ces sorties sont des vecteurs de lien de connexion qui ne donnent aucune indication supplémentaire sur la contribution des variables lors de la classification supervisée.
— *Le modèle SVM* a pour objectif de trouver l'hyperplan optimal qui sépare au mieux les données dans l'espace d'entré. Cependant, les seules informations fournies par ce dernier sont en général soit les vecteurs de support sans aucune autre information, soit les coefficients de l'hyperplan de séparation et éventuellement le taux de bonnes classifications. L'utilisateur trouve donc une difficulté d'expliquer ce qui fait qu'un individu est dans une classe plutôt que dans une autre.

Pour une interprétation aisée des résultats fournis par ces modèles, plusieurs techniques ont vu le jour. Les techniques proposées peuvent être soit dédiées ou généralistes. Dans le premier cas, les techniques sont fondées sur le fonctionnement interne du modèle en question. Par conséquent, ces techniques ne peuvent être utilisées que pour ce modèle. Au contrario, les techniques généralistes sont des techniques utilisables pour tous les modèles de classification. Le tableau 2.3 présente quelques méthodes de sélection des variables généralisées et dédiées permettant de faciliter la tâche de l'interprétation des résultats issus des ANN et des SVM .

Les méthodes dédiées					les méthodes généralisées			
Méthodes	classifieur	Année	Techniques	Source	Méthodes	Année	Techniques	Source
SVM-RFE	SVM	2002	embedded	[54]	Robnik et al.	2008	filter	[90]
Féraud et al.	ANN	2002	embedded	[47]	Oh et al.	2004	embedded	[83]
MOI	SVM	2004	wrapper	[94]	Penget al.	2005	filter	[84]

TABLE 2.3 – Techniques de sélection des variables pour les ANN et les SVM

La technique d'extraction des règles est également l'une des techniques permettant de résoudre la problématique d'interprétation des résultats générés par les modèles boîtes noires. Le principe de cette technique est d'extraire un nombre réduit de règles qui imitent le fonctionnement des modèles opaques. Elle peut être une technique dédiée si l'approche suivie est «décompositionnelle» et elle peut être généraliste si l'approche suivie est «pédagogique». Les deux notions «décompositionnelle» et «pédagogique» seront discutées dans ce qui suit.

L'importance de la technique d'extraction des règles réside essentiellement dans le fait qu'elle :

1. Fournit un nombre réduit de règles qui imitent *fidèlement* le comportement du modèle boîte noire en termes de prédiction. Cet ensemble de règle est souvent compréhensible par l'utilisateur.
2. Améliore les performances des techniques d'induction des règles en supprimant par exemple le bruit présent dans les données. Cela peut être fait en remplaçant la variable cible par les prédictions faites par le modèle opaque. On constate donc qu'un modèle boîte noire performant peut être utilisé dans une étape de prétraitement pour nettoyer les données [75].
3. Étend l'utilisation du modèle *boîte noire* à des domaines "critiques" (*i.e.*, les domaines où l'interprétation est un critère crucial comme par exemple la médecine).

Lors de la construction d'un algorithme d'extraction des règles, plusieurs questions peuvent apparaitre :

— *Quelle logique faut-il suivre pour former les règles ?*
— *A quel niveau d'apprentissage faut-il extraire les règles ?*
— *Comment mesurer la cohérence entre les règles extraites et les prédictions faites par les modèles opaques ?*
— *... etc.*

Pour répondre à ces différentes questions, Andrews et al. [14] ont proposé un système de classification (basé sur cinq critères) pour l'extraction des règles à partir des réseaux de neurones. Ce système peut être peut être étendu à d'autres modèles opaques tels que les SVM ; Ces critères sont : (1) la transparence de l'algorithme d'extraction par rapport au modèle sous-jacent, (2) la puissance expressive des règles, (3) la qualité des règles extraites, (4) la scalabilité de l'algorithmique et (5) la consistance de l'algorithme.

(1) La transparence de l'algorithme par rapport au modèle : Ce critère définit la relation entre les algorithmes d'extraction des règles et l'architecture interne du modèle sous-jacent. Selon la taxonomie présentée par Andrews et al. [14], il existe trois manières différentes pour

extraire les règles à partir d'un modèle boîte noire à savoir les techniques *décompositionnelles*, les techniques *pédagogiques* et les techniques *éclectiques* :

— *Les techniques décompositionnelles* sont étroitement liées au fonctionnement interne du modèle sous-jacent. Par exemple, pour les réseaux de neurones, ces techniques s'intéressent au fonctionnement interne de chaque neurone du réseau. Ensuite, les règles extraites sont agrégées afin d'avoir une relation globale. Pour les SVM, ces techniques utilisent une approximation locale de la frontière de décision en utilisant des hyper-rectangles ou des ellipsoïdes comme régions dans l'espace d'entrée (voir Figure 2.9 (b)).

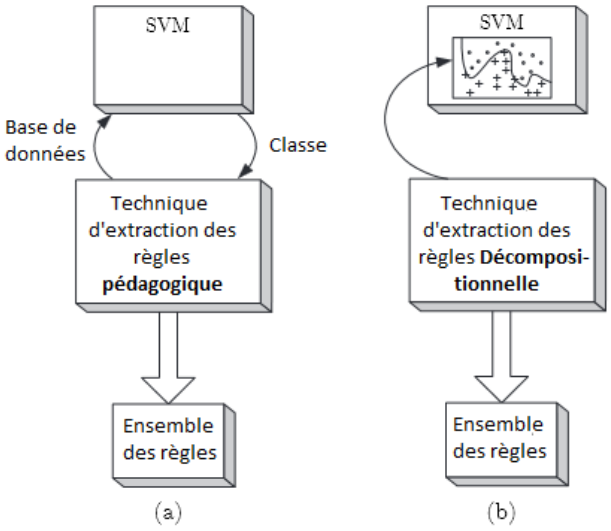


FIGURE 2.9 – Exemple illustratif de la technique décompositionnelle (b) et la technique pédagogique (a) dans le cadre des SVM

— *Les techniques pédagogiques* Les algorithmes qui suivent la technique pédagogique ne sont pas des algorithmes d'extraction des règles au sens strict du mot. En effet, ils extraient directement les connaissances utiles à partir des données d'apprentissage. Pour interpréter les résultats du modèle opaque en utilisant les algorithmes qui suivent cette technique, il suffit donc de remplacer la variable cible par les prédictions générées par celui-ci. À titre d'exemple, voir la figure 2.9 (a) dans le cadre des SVM.

— *Les techniques éclectiques* combinent les deux techniques décompositionnelle et pédagogique à la fois.

(2) La puissance expressive des règles : dépend généralement de la langue utilisée par l'utilisateur pour les exprimer. Autrement dit, elle dépend de la forme de celles-ci. Les règles les plus utilisées dans la littérature, à notre connaissance, sont les règles *propositionnelles*, les règles *logiques de forme* ***M*** *de* ***N*** *conditions*, les règles *floues*, les règles *obliques* et les règles *équations* :

— *Les règles propositionnelles* sont les règles les plus répandues dans la littérature en raison de leur simplicité. Ces règles prennent la forme suivante : "Si *condition* Alors *expression*". Dans le cadre d'extraction des connaissances, la plupart des algorithmes veilleront à ce que les règles soient mutuellement exclusives dans le but de ne pouvoir utiliser qu'une seule règle pour la prise de décision quand une nouvelle observation est présentée. Néanmoins,

il existe aussi des algorithmes qui permettent d'extraire plusieurs règles pour une seule observation, ce qui nécessite l'utilisation d'un mécanisme supplémentaire pour combiner les différentes prédictions. Par exemple, dans [33], Chen associe un facteur de confiance à chacune des règles de telle sorte que les règles tirées avec une grande confiance auraient un grand impact sur la décision finale.

— *Les règles logiques de forme M de N conditions* sont les règles de la forme : "Si *au moins* M de N conditions Alors *expression*" où M est un entier et N est un ensemble de conditions. Ces règles sont étroitement liées aux règles propositionnelles puisqu'elles peuvent facilement être transformées en celles-ci.

— *Les règles floues* sont les règles qui prennent la forme suivante : " Si X est faible et Y est moyen Alors *expression*". Généralement, les règles floues sont à la fois compréhensibles et faciles puisqu'elles sont exprimées par des concepts linguistiques faciles à interpréter par l'utilisateur.

— *Les règles obliques* sont basées sur des fonctions discriminantes par morceaux : "Si $(\alpha x_1 + \beta x_2 > c_1)$ et $(\sigma x_1 + \rho x_2 > c_2)$ et ... Alors *expression*". Ces règles sont généralement plus difficiles à comprendre par rapport aux règles propositionnelles. Toutefois, elles se caractérisent par leurs capacités à créer des frontières qui ne sont pas forcément parallèles aux axes de l'espace d'origine d'entrée. Par conséquent, elles nécessitent moins de conditions que les règles propositionnelles.

— *Les règles équations* sont les règles qui contiennent une fonction polynomiale dans la partie condition : "Si $\alpha x_1^2 + \beta x_2^2 + \sigma x_1 x_2 + \rho x_1 + \varphi x_2$ Alors *expression*". La façon dont ces règles ont été construites les rend plus difficiles à comprendre et par conséquent elles contribuent peu à l'interprétation des modèles opaques.

(3) La qualité des règles extraites : Andrews et al.[14] ont proposé un ensemble de trois critères pour évaluer la qualité des règles à savoir : *La précision*, *la fidélité* et *la compréhensibilité* :

— *La précision* mesure la capacité des règles extraites à prédire correctement les classes des nouvelles instances dans l'ensemble des données de test. Elle est définie généralement comme le pourcentage des instances bien classées.

— *La fidélité* est étroitement liée à la précision. Elle mesure la capacité des règles à imiter la prédiction du modèle d'apprentissage à partir duquel elles ont été extraites.

— *La compréhensibilité* est mesurée par le nombre des règles extraites et le nombre des antécédents par règle (i.e. nombre des variables).

(4) La scalabilité de l'algorithmique : La scalabilité définit généralement la capacité de l'algorithme à faire face aux problèmes de grandes dimensions (un très grand nombre de variables d'entrées) et/ou de grande taille (nombre d'exemples élevé) de la même façon que les problèmes jouets. Bien évidemment, ce critère dépend du temps d'exécution et de la performance de l'algorithme. Cependant, dans le cadre d'extraction des règles, à côté du temps d'exécution de l'algorithme, les règles extraites devraient rester compréhensibles quel que soit la dimension ou la taille de l'ensemble d'apprentissage. La scalabilité mentionne la façon dont le temps d'exécution de l'algorithme et la compréhensibilité des règles extraites varient en fonction de différents facteurs tels que le modèle opaque, la taille de l'ensemble d'apprentissage et le nombre des variables d'entrées [36].

(5) La consistance de l'algorithme : La consistance d'un algorithme peut prendre plusieurs définitions ; Par exemple, elle peut être définie comme étant la capacité à générer, sous différentes sessions d'apprentissage, des règles avec les mêmes degrés d'accuracy. En outre, dans

[63], Johansson *et al.* définissent la consistance de l'algorithme comme étant sa capacité à extraire des règles similaires à chaque fois qu'il est appliqué à un même ensemble de données. Cependant, les auteurs soulignent immédiatement la difficulté associée à cette définition, puisqu'il n'y a pas de définition simple de similitude des règles (à notre connaissance).

Les tableaux 2.4 et 2.5 présentent respectivement quelques méthodes d'extraction des règles (décompositionnelle et pédagogique) permettant de faciliter la tâche d'interprétation des résultats issus des ANN et des SVM .

Les réseaux de neurones (ANN)				
Méthode	**Année**	**Technique**	**Type**	**Source**
RN2	2002	Crée des règles de forme polynomiale, RN2 garantit de produire des unités dans les unités cachées et regroupe les valeurs d'activation des unités cachées en utilisant un algorithme de clustering.	D	[92]
REFANN	2002	Approxime la fonction d'activation des ANN par des fonctions linéaires par morceaux	D	[93]
GEX	2004	Algorithme génétique, fournit des règles propositionnelles	P	[74]
BUR	2004	Basé sur " Gradient Boosting Machines"	P	[33]
ITER	2006	Basé sur une augmentation itérative des hypercubes	P	[58]
Coalition-opposition	2007	Extrait des coalitions et des oppositions minimaux du neurone à partir d'un arbre.	D	[16]

TABLE 2.4 – Techniques décompositionnelles (D) et pédagogiques (P) d'extraction des règles à partir des ANN

Les machines à vecteur de support (SVM)				
Méthode	**Année**	**Technique**	**Type**	**Source**
CART	1984	Arbre de décision	P	[25]
CN2	1989	Induction par règles, algorithme de recouvrement séquentiel	P	[34]
C4.5	1993	Arbre de décision	P	[87]
SVM + prototype	2002	Clustering, basé sur des régions ellipsoïdes et hyper-rectangulaires	D	[79]
RulExSVM	2004	Fournit des règles propositionnelles, cet algorithme est basé sur des régions hyper-rectangulaires	D	[103]
HRE	2005	SVC (Support Vector Clustering), basé sur des règles hyper-rectangulaires	D	[104]
Fung et al.	2005	Algorithme itératif pour les SVM linéaire, basé sur une programmation linéaire	D	[51]
Hien et al.	2014	Fournit des règles floues	D	[80]

TABLE 2.5 – Techniques décompositionnelles (D) et pédagogiques (P) d'extraction des règles à partir des SVM

2.4 Interprétation

L'interprétation des résultats issus d'un modèle (descriptif ou prédictif) est un acte très subjectif. Il dépend généralement de l'utilisateur et du domaine d'application. En effet, les connaissances utiles doivent être exprimées dans la langue et la sémantique de celui-ci. De plus, l'interprétation des résultats du modèle diffère d'un domaine d'application à l'autre en fonction du niveau de détails demandé (voir Section 2.4.3).

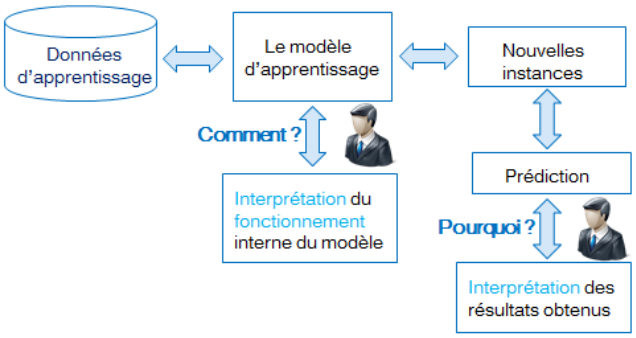


FIGURE 2.10 – Emplacement des deux types d'interprétation au cours du processus de la classification

Dans le cadre de l'apprentissage automatique, deux types d'interprétation se distinguent : l'interprétation du modèle et l'interprétation des résultats issus de ce dernier. Dans le premier cas, l'utilisateur cherche à expliquer comment le modèle fonctionne afin d'obtenir ses résultats. Il cherche à comprendre la logique suivie par le modèle pour atteindre l'objectif désiré. Dans le deuxième cas, l'utilisateur s'intéresse à expliquer pourquoi un tel résultat a été obtenu par le modèle. Il s'agit donc de connaître les différents facteurs qui ont un grand impact sur l'obtention de ce résultat. A titre d'exemple, l'interprétation des machines à vecteurs de support (SVM) se fait à travers la compréhension de son fonctionnement interne (e.g l'obtention des vecteurs supports, de l'hyperplan de séparation des classes, etc.). L'interprétation des résultats des SVM, quant à elle, se fait à l'aide de la compréhension des relations présentes entre les entrées et la sortie de l'algorithme. Il s'agit alors de détecter les causes des prédictions (voir Section 2.4.1). La figure 2.10 illustre schématiquement l'emplacement de ces deux types d'interprétation au cours du processus de classification.

Généralement, on dit d'un modèle qu'il est interprétable s'il est capable de fournir à l'utilisateur des résultats compréhensibles : si, à partir des résultats obtenus par le modèle, l'utilisateur peut extraire facilement les connaissances utiles en se basant sur les variables natives. A titre d'exemple :

1. **Dans le cadre du clustering** : On dit qu'un modèle est interprétable s'il permet à l'utilisateur de comprendre les causes de la formation des clusters. On parle ici d'une interprétation locale (voir Section 2.4.3). Il s'agit donc d'identifier les variables qui contribuent le plus à la formation de chaque cluster. En effet, certaines variables peuvent être discriminantes pour la formation d'un cluster et s'avérer peu révélatrices pour la formation d'autres. Dans le cas où on s'intéresse au traitement global du processus de clustering, la description des clusters n'est pas nécessaire. Seule une analyse de qualité peut être

suffisante. On parle d'ici d'une interprétation globale (voir Section 2.4.3). Pour plus de détails sur l'interprétation des résultats de clustering voir Section 2.2.3.

2. **Dans le cadre de la classification** : On dit qu'un modèle est interprétable s'il permet de répondre aux deux questions suivantes : (1) Quelles sont les causes des prédictions ? (2) Quelle est la fiabilité d'une prédiction ?. Les réponses à ces deux questions sont présentées respectivement dans les deux sections 2.4.1 et 2.4.2.

2.4.1 Les raisons d'une prédiction

Intuitivement, la façon qui permet de comprendre les causes d'une prédiction est de réaliser à quel point les variables ont contribué au résultat du modèle. Généralement, les données d'apprentissage dont on dispose ne contiennent pas forcément que des variables pertinentes. Il est possible que certaines variables correspondent à du bruit ou qu'elles soient corrélées, redondantes, peu informatives ou même inutiles au problème de la classification. Les variables pertinentes sont souvent celles qui ont un fort impact sur la prédiction par rapport à celles non informatives. De ce fait, on cherchera à **"mesurer l'importance des variables"** en fonction de leur contribution dans le processus d'apprentissage.

Il est à noter que la mesure d'importance des variables peut être réalisée à plusieurs niveaux : avant, après ou au cours du processus d'apprentissage.

— *Avant l'apprentissage* : les informations disponibles à ce stade sont seulement les données. Cette technique est indépendance du modèle sous-jacent. L'utilisateur cherche donc à identifier les variables les plus importantes en s'appuyant sur les propriétés des données. C'est-à-dire celles qui séparent aux mieux les données.
— *Au cours de l'apprentissage* : Cette technique est étroitement liée au modèle d'apprentissage. Elle intègre les performances prédictives du classifieur dans la procédure du calcul d'importance des variables. La mesure de cette importance s'effectue dans ce cas suivant une procédure itérative : elle compare plusieurs résultats issus du modèle en utilisant le même jeu de données afin d'évaluer l'importance des variables.
— *Après l'apprentissage* : s'appuie sur une connaissance *a priori* des caractéristiques d'un bon modèle et du résultat obtenu par ce dernier. A ce stade, le résultat final généré par le modèle est considéré comme une référence afin de mesurer l'importance des variables.

Le calcul de l'importance d'une variable permet de mesurer à quel point l'information qu'elle contient a été décisive dans l'obtention du résultat. Dans le cas où l'on dispose d'un nombre important de variables ou lorsqu'on utilise un modèle boîte noire (voir Section 2.3.2), plusieurs techniques ont vu le jour permettant ainsi une interprétation aisée des résultats. A travers l'utilisation de ces techniques, l'utilisateur peut concentrer son attention sur les variables les plus pertinentes :

• **Le tri des variables en fonction de leur importance** consiste à classer les variables en fonction de leur contribution à la formation : *i*) des groupes d'instances homogènes si on est dans le cadre du clustering ou *ii*) des groupes d'instances ayant la même classe si on est dans le cadre de la classification supervisée. Cette technique permet à l'utilisateur de se concentrer sur les variables importantes pour une interprétation aisée. L'importance des variables peut être mesurée à l'aide de plusieurs critères. A titre d'exemple, on citera l'information mutuelle et le critère de Fisher pour la classification supervisée et l'indice Davies Bouldin et la Silhouette pour le clustering.

• **La pondération** consiste à affecter des poids aux variables qui évoluent au cours de l'apprentissage en fonction de leur importance. Il s'agit donc de donner un rôle plus important

aux variables contenant l'information intéressante pendant l'apprentissage. Ces poids servent ensuite lors de la phase d'interprétation des résultats.

• **La sélection des variables** a pour objectif de trouver un sous-ensemble *pertinent* de variables parmi celles de l'ensemble de départ. L'avantage de cette méthode réside dans le fait qu'elle permet : (1) d'améliorer souvent la performance prédictive du modèle, (2) de faciliter l'interprétation des résultats, et (3) d'étendre l'utilisation des modèles *boîtes noires* à des domaines "critiques". Le processus de sélection des variables passe généralement par trois étapes différentes, à savoir : (1) un algorithme de recherche, (2) un critère d'évaluation, et (3) un critère d'arrêt.

(1) L'algorithme de recherche a pour objectif d'explorer l'espace de combinaison des variables. Il peut être *Exhaustif*, *Heuristique* ou *Aléatoire* :

— *Exhaustif* : Cette catégorie consiste à sélectionner le meilleur sous-ensemble des variables parmi tous les sous-ensembles existant en faisant une recherche exhaustive. Cependant, le problème majeur de cette stratégie est que le nombre de combinaisons des variables possibles croit exponentiellement quand le nombre des variables augmentent. Ceci rend la recherche exhaustive quasiment impossible.

— *Heuristique* : Les algorithmes qui appartiennent à cette catégorie sont généralement les algorithmes itératifs ; A chaque itération, une ou des variables peuvent être ajoutées ou rejetées selon leurs importances. Ces algorithmes sont connus généralement par leur simplicité et leur rapidité. Dans la littérature, cette catégorie se divise en trois types de procédure de recherche à savoir *Forward*, *Backward* et *Stepwise*.

1. *Forward* : L'objectif de cette procédure est de partir d'un ensemble vide de variables et d'ajouter successivement, à chaque itération, une ou des variables pertinentes.
2. *Backward* : Le principe de cette procédure de recherche est de partir de l'ensemble global de toutes les variables et de supprimer séquentiellement, à chaque itération, une ou des variables (les moins pertinentes).
3. *Stepwise* : L'idée centrale de cette approche est d'ajouter ou de rejeter une ou des variables au sous-ensemble de variables courant.

— *Aléatoire* : L'idée derrière cette catégorie est de générer aléatoirement un nombre de sous-ensembles de variables afin de sélectionner le 'meilleur' sous-ensemble parmi ces derniers. Cette catégorie est appelée aussi, l'approche stochastique.

(2) Le critère d'évaluation permet de mesurer la qualité d'un sous-ensemble de variables suivant que l'on utilise l'approche *filter*, *wrapper* ou *embedded* :

— *L'approche 'filter'* : Cette approche est indépendante du modèle sous-jacent. Autrement dit, elle présélectionne les variables, puis elle utilise ces variables sélectionnées dans le modèle d'apprentissage. Cependant, cette approche peut être considérée comme une étape de prétraitement (avant la phase d'apprentissage). L'approche *filter* repose sur le calcul d'un score qui permet de calculer la pertinence de chaque variable en s'appuyant sur les propriétés des données d'apprentissage. Plusieurs scores ont été proposés dans la littérature permettant ainsi de mesurer la pertinence d'une variable. Ces scores peuvent aussi être utilisés comme un critère d'évaluation. Parmi ces scores, on trouve *le critère de Fisher*, *le critère de corrélation* et *l'information mutuelle*.

— *L'approche 'wrapper'* : Contrairement à l'approche *filter*, les méthodes appartenant à l'approche *wrapper* intègrent les performances prédictives du classifieur dans la procédure de recherche pour évaluer la qualité des sous-ensembles de variables.

— *L'approche 'embedded'* : Le principe de cette approche est d'incorporer la sélection lors du processus d'apprentissage. La sélection des variables s'appuie donc sur un critère propre à la méthode. Les arbres de décision sont l'illustration la plus emblématique. Dans les méthodes de sélection de type "*wrapper*", la base d'apprentissage est divisée en deux parties : une base d'apprentissage et une base de validation pour valider le sous-ensemble de caractéristiques sélectionné. En revanche, les méthodes "*embedded*" peuvent se servir de tous les exemples d'apprentissage pour établir le système.

(3) Les critères d'arrêt les plus utilisés, pour les trois approches cités ci-dessus ('*filter*', '*wrapper*' et '*embedded*') sont :

— *Pour l'approche filter* : Une fois les variables triées selon leur importance dans le processus, l'utilisateur peut sélectionner les variables les plus pertinentes afin de les utiliser par un classifieur. Le nombre de variables sélectionné est donc fixé a priori par celui-ci.

— *Pour l'approche wrapper* : Le critère d'arrêt est basé sur une fonction d'évaluation et dépend de deux faits : *i*) l'ajout ou la suppression d'une variable ne produit aucun sous ensemble plus performant, et *ii*) le sous ensemble obtenu est, d'après la fonction d'évaluation, le sous ensemble optimal. Le processus continue jusqu'à ce que le critère d'arrêt soit satisfait.

— *Pour l'approche embedded* : Le processus de recherche s'arrête lorsque la précision dépasse un certain seuil fixé a priori par l'utilisateur.

2.4.2 La fiabilité d'une prédiction

Les prédictions générées par les modèles d'apprentissage sont souvent susceptibles de contenir des erreurs. Les critères permettant de mesurer la qualité prédictive des modèles d'apprentissage (e.g., l'accuracy) fournissent souvent à l'utilisateur une information "globale" sur leurs capacités à bien classer des nouvelles instances. Néanmoins, ces critères ne fournissent pas une information "locale" sur l'erreur de prédiction prévue pour chaque instance en particulier. D'un autre côté, les modèles les plus performants comme par exemple les SVM sont des modèles complexes qui ne permettent pas à l'utilisateur de comprendre facilement pourquoi une prédiction particulière a été faite. À partir de ce constat, on trouve qu'il est utile que l'utilisateur soit informé de la fiabilité de chaque prédiction.

Pour répondre à cette problématique, il semble important de comprendre la relation entre les variables descriptives et la valeur prédite. Plus précisément, il s'agit ici de détecter l'effet du changement de la valeur de chaque variable sur la valeur prédite. La mesure de tel effet permet d'une part de connaître l'importance de chaque variable et d'autre part de déterminer les valeurs qui sont les seuils du changement de la valeur prédite. Plusieurs travaux ont été effectués dans ce cadre, on pourra citer notamment ceux proposés par Robnik *et al.* [90] et par Briesemeister *et al.* [27].

2.4.3 La granularité d'une interprétation

Selon le besoin de l'utilisateur et du domaine d'application auquel il s'intéresse, le concept d'interprétation peut prendre plusieurs formes : *individuelle, locale et globale.*

Interprétation individuelle : Dans ce cas, on s'intéresse à l'interprétation de la prédiction prévue pour une instance en particulier. Il s'agit donc de découvrir les différents facteurs qui influent la sortie de cette instance. Pour atteindre cet objectif, l'utilisateur doit donc déterminer

la relation entre les valeurs des variables descriptives et la valeur prédite pour cette instance. Ceci peut être réalisé à titre d'exemple : *i*) en mesurant l'importance des variables selon leur contribution à l'obtention de cette estimation. *ii*) en déterminant la fiabilité de cette prédiction individuelle (voir Section 2.4.2), *etc.*. Ce genre d'interprétation est très utile dans plusieurs domaines d'application. Par exemple, dans le domaine de la finance : lors de la prise de la décision d'approbation des crédits dans les banques, l'agent doit être capable d'expliquer au client les raisons du refus d'un crédit.

Interprétation locale : Dans ce cas, on s'intéresse à la description d'un groupe en particulier. Il s'agit ici de comprendre pourquoi les instances appartenant à un même groupe ont la même sortie. Ceci revient à réaliser à quel point les valeurs de chaque variable influent sur la décision prédictive du groupe en question. Que ce soit dans le cadre de la classification supervisée ou le cadre du clustering, plusieurs techniques ont été proposées pour atteindre cet objectif. Parmi ces méthodes, on trouve l'importance des variables, la pondération, la sélection des variables, *etc.* Ce genre d'interprétation est très utile dans plusieurs domaines d'application. Par exemple, dans le cadre de la Gestion de la Relation Client, l'agent doit connaître les besoins et les attentes des clients en fonction de leurs comportements afin qu'il puisse adapter les actions marketing vers ces clients.

Interprétation globale : Dans ce cas, la description des groupes ou d'une instance en particulier n'est pas nécessaire. Seul un traitement global des résultats est suffisant. L'utilisateur cherche : *i*) soit à découvrir l'ensemble des facteurs qui influent les décisions prises. On parle ici de la contribution des variables dans l'obtention des résultats. *ii*) soit à orienter le traitement vers l'analyse globale de la qualité des résultats obtenus. Cette qualité dépend de l'apprentissage suivi. Par exemple, dans le cadre de la classification supervisée, les critères de qualité les plus utilisés sont : l'accuracy (ACC), AUC, la courbe de Lift, l'ARI (Adjusted Rand Index),..., *etc.* Dans le cadre du clustering, les critères de qualité les plus utilisés sont : le critère MSE, l'inertie intra\inter clusters, la Silhouette [91], l'indice SD [55], l'indice Davies-Bouldin [38],..., *etc.*

Le tableau 2.6 présente les différents moyens permettant d'aboutir à ces formes d'interprétation.

	Classification supervisée			Clustering			
	Importance	Sélection	Qualité	Importance	Sélection	Qualité	Profil moyen
individuelle	✓	✓	–	✓	✓	–	–
Locale	✓	✓	–	✓	✓	–	✓
Globale	✓	✓	✓	✓	✓	✓	✓

TABLE 2.6 – Techniques permettant de réaliser les différents types d'interprétation

2.5 Approche descriptive et prédictive simultanément

2.5.1 Contexte

Depuis quelques années, les chercheurs ont concentré leur attention sur l'étude d'un nouvel aspect d'apprentissage. Ce dernier fusionne à la fois les caractéristiques de la classification supervisée (la prédiction) et du clustering (la description). Les algorithmes appartenant à ce type

d'apprentissage cherchent à décrire et de prédire d'une manière simultanée. Autrement dit, ces algorithmes visent à découvrir la structure interne de la variable cible. Puis, munis de cette structure, ils cherchent à prédire la classe des nouvelles instances.

Dans le domaine de l'apprentissage automatique, il existe principalement deux axes à traiter, à savoir l'axe de prédiction et l'axe de description. Dans le premier axe, on cherche à prédire une valeur (pour la régression) ou une classe (pour la classification supervisée) pour une nouvelle donnée à partir d'un ensemble de données d'apprentissage (voir Section 2.3). Contrairement au deuxième axe où l'on cherche à découvrir la structure sous-jacente d'un ensemble de données (voir Section 2.2) à partir de la distinction des groupes d'instances homogènes.

À côté de ces deux axes, on peut ajouter un troisième axe qui est l'axe d'interprétation (voir Section 2.4). Dans certains domaines appelés "domaines critiques" (*e.g.,* la médecine, les services bancaires, *etc*), la compréhension (ou la description) des résultats issus d'un modèle d'apprentissage est une condition aussi importante que sa performance prédictive. Dans ces domaines, l'utilisateur doit avoir un certain niveau de confiance vis-à-vis des hypothèses générées par le modèle. L'accident nucléaire de Three Mile Island [1] est l'un des exemples concrets qui montre la nécessité d'utiliser un modèle à la fois performant et interprétable. Le facteur majeur derrière cet incident est que la personne n'a pas eu confiance dans les recommandations faites par la machine. Le domaine médical est aussi l'un des domaines critiques où la vie de l'être humain est mise en jeu. Lors de la prise d'une décision en se basant sur un modèle d'apprentissage, le médecin doit être précis et convaincant. Par exemple, si cette décision conduit à un préjudice majeur pour le patient, alors le médecin doit être capable de défendre sa décision s'il est accusé de négligence médicale. Dans de telles situation, la qualité des modèles utilisés réside dans leurs capacités à fournir des résultats étant à la fois performants en prédiction et compréhensibles.

Les algorithmes de la classification supervisée traitent principalement l'axe de prédiction. Le but majeur de ces algorithmes est d'apprendre, à partir d'un ensemble de données, un modèle permettant de prédire ultérieurement la classe de nouvelles instances. Pour certains algorithmes de la classification supervisée, ce processus est effectué sans prendre en considération la probabilité d'existence d'une structure sous-jacente au sein d'au moins une des classes (*i.e.,* la description). À titre d'exemple, le jeu de données présenté dans la figure 2.11 est caractérisé par la présence de trois sous-groupes différents pour la classe Virginica et de deux sous-groupes différents pour la classe Setosa.

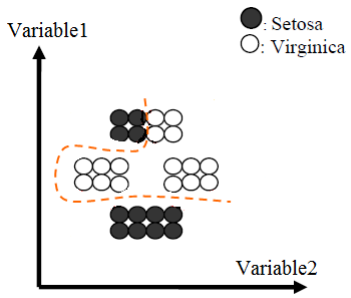


FIGURE 2.11 – Principe de la classification supervisée via les SVM

Pour cet exemple illustratif, un algorithme d'apprentissage tel que les machines à vecteurs de support (SVM) va construire une frontière de décision non linéaire séparant les deux classes sans accorder la moindre importance à la structure interne de la variable cible. On déduit que

1. http ://en.wikipedia.org/wiki/Three_Mile_Island_accident

certains d'algorithmes de la classification supervisée peuvent avoir du mal à décrire l'ensemble des données. Concernant l'axe d'interprétation, les algorithmes de la classification supervisée peuvent être divisés en deux grandes catégories. La première catégorie englobe l'ensemble des algorithmes d'apprentissage performants mais qui fournissent des résultats difficilement immédiatement compréhensibles par l'utilisateur. C'est le cas des modèles appelés "boîtes noires" (*e.g.,* les ANN et les SVM, voir Section 2.3.2). La deuxième catégorie, quant-à-elle, englobe l'ensemble des algorithmes d'apprentissages qui sont naturellement plus interprétables. Ces derniers sont souvent des algorithmes moins performants par rapport aux modèles boîtes noires. C'est le cas des modèles transparents (*e.g.,* les arbres de décision, voir Section 2.3.1).

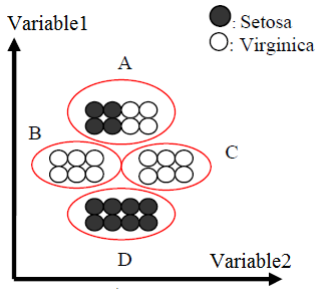


FIGURE 2.12 – Principe du clustering

Les algorithmes de clustering, quant à eux, traitent principalement l'axe de description. En effet, ces algorithmes cherchent à subdiviser l'ensemble des données en un certain nombre de groupes (ou clusters) de manière à ce que les instances soient similaires au sein de chaque groupe et dissimilaires d'un groupe à l'autre. Cette notion de similarité/dissimilarité entre les individus est définie dans le cadre non supervisé (c'est-à-dire, l'absence d'une classe à prédire). Cependant, deux instances proches en termes de distance peuvent appartenir à des classes différentes. Par exemple, le groupe A de la partie gauche de la figure 2.12. A partir de ce constat, on déduit que les algorithmes classiques de clustering peuvent avoir du mal à prédire la classe des nouvelles instances.

À partir de ce constat, on déduit que les algorithmes de la classification supervisée et du clustering ont du mal à décrire et à prédire d'une manière simultanée sous la contrainte d'interprétation. Pour tenter de résoudre cette problématique, deux grandes voies existent (voir la figure 2.13).

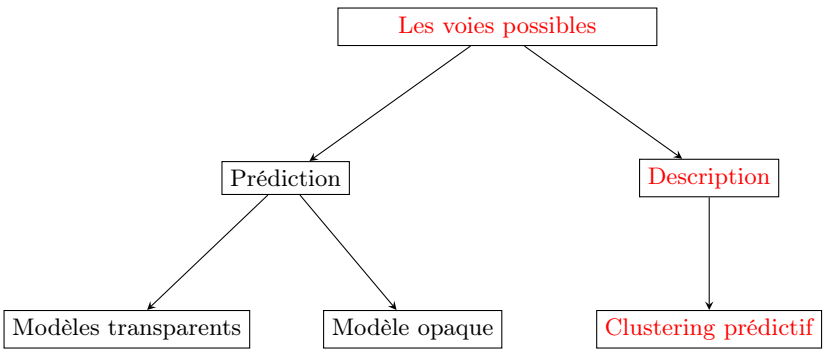


FIGURE 2.13 – Les différentes voies possibles pour la description et la prédiction simultanées

La première voie consiste à adapter les algorithmes de la classification supervisée au problème de la description des données. Dans ce cas, on se trouve face à deux possibilités :

1. rendre les algorithmes transparents plus performants en prédiction et en description sans perdre leur faculté d'intelligibilité des résultats.
2. modifier les modèles boîtes noires pour permettre une bonne description des données. Ces algorithmes doivent posséder également une technique permettant une interprétation facile des résultats.

La deuxième voie, quant-à-elle, consiste à rendre les algorithmes de clustering plus performant en prédiction sans perdre leur faculté à bien décrire les données. Ce type d'apprentissage est appelé *le clustering prédictif*. Dans cette thèse, nous nous intéressons exclusivement à la problématique traitée par cette deuxième voie.

2.5.2 Clustering prédictif

A. Définition

Le clustering prédictif englobe principalement l'ensemble des algorithmes de clustering soumise à des modifications dans le but de les adapter au problème de la classification supervisée. Ceci est effectué en préservant la faculté de l'algorithme à bien décrire les données. L'objectif majeur des algorithmes du clustering prédictif est de découvrir dans la phase d'apprentissage la structure interne de la variable cible. Puis, munie de cette structure, ces algorithmes cherchent à prédire la classe des nouvelles instances. Dans la littérature, il existe deux grandes catégories du clustering prédictif (voir les deux figures 2.14 et 2.15). **La première catégorie** privilège l'axe de prédiction par rapport aux deux autres axes (*i.e.,* l'interprétation et la description) tout en exigeant de minimiser le nombre de groupes appris dans la phase d'apprentissage (voir la figure 2.14). Par contre, les algorithmes de **la deuxième catégorie** cherchent dans la phase d'apprentissage à réaliser le compromis entre la description et la prédiction en découvrant la structure interne *complète* de la variable cible (voir la figure 2.15).

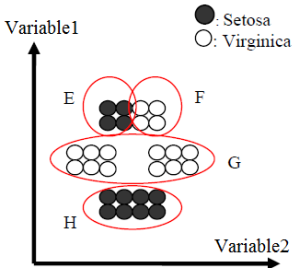


FIGURE 2.14 – Premier type du clustering prédictif

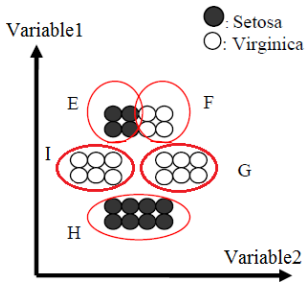


FIGURE 2.15 – Deuxième type du clustering prédictif

B. État de l'art

Dans la littérature, il existe plusieurs variations du clustering prédictif, à savoir, la décomposition des classes, l'arbre de décision, le clustering supervisé, le clustering prédictif de Dzeroski *et al.* et le clusterwise.

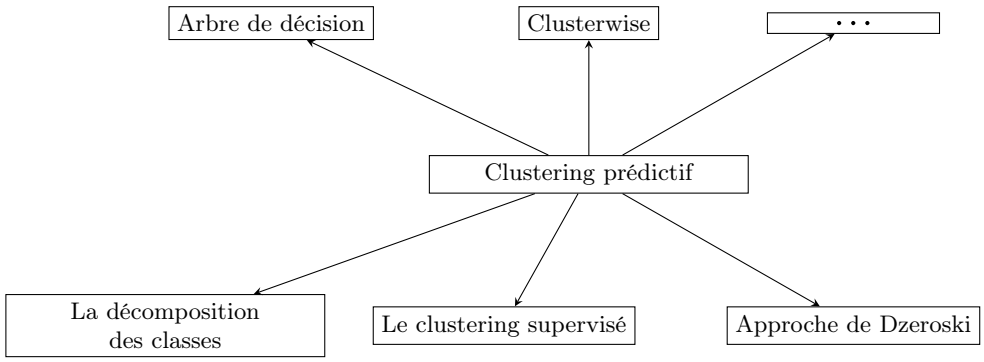


FIGURE 2.16 – Les différentes variations du clustering prédictif

B.1. la décomposition des classes

La technique de décomposition des classes consiste à décrire chaque classe individuellement en utilisant un algorithme de clustering. Chaque classe est donc décrite par un certain nombre de clusters. Ce nombre peut être différent d'une classe à l'autre. À la fin de la phase d'apprentissage, on obtient alors P clusters résultants ($P = \sum_{j=1}^{J} K_j$) avec J est le nombre de classes dans le jeu de données. Dans la littérature, la décomposition des classes est utilisée souvent pour améliorer la performance des classifieurs linéaires simples comme les SVM linéaires (*e.g.,* Vilalta et al. et Wu et al.). Cette technique se résume en deux étapes principales : (1) réalisation d'un clustering de type k-moyennes (où K_j est selon les auteurs une entrée ou une sortie de l'algorithme) par groupe d'exemples qui appartiennent à la même classe j, (2) entraînement d'un classifieur sur les P classes résultantes et interprétation des résultats.

La technique de la décomposition des classes engendre dans la phase d'apprentissage des groupes totalement purs en termes de classes (traitement individuel de chaque classe). De plus, chaque groupe formé contient probablement les instances les plus homogènes possibles et qui diffèrent des instances des autres groupes en raison de l'utilisation d'un algorithme de clustering pour décrire les classes. Cependant, dans le cadre du clustering prédictif, cette technique risque de générer, dans la phase de test, des prototypes (si un algorithme de partitionnement est utilisé) virtuels en raison de forte proximité entre deux prototypes de classes différentes dans la phase d'apprentissage. À titre d'exemple, si le jeu de données contient du bruit comme illustré dans la figure 2.17, les clusters de différentes classes formés dans la phase d'apprentissage peuvent être proches les uns des autres (par exemple, les deux clusters A et B de la figure 2.17). Par conséquent, la probabilité d'affecter les nouvelles instances à seulement l'un des clusters est importante.

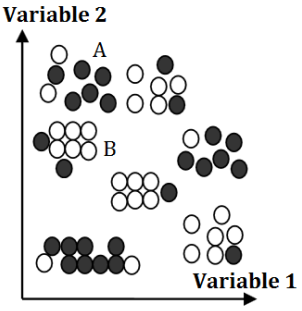


FIGURE 2.17 – Problème d'une classification binaire

B.2. Les arbres de décision

Un arbre de décision peut être considéré comme une hiérarchie de clusters, où chaque nœud représente un cluster. Un tel arbre est appelé un arbre de clustering. Sa structure récursive contient une combinaison de nœuds et de feuilles internes (Figure 2.18). Chaque nœud spécifie un test à effectuer sur une seule variable et ses branches indiquent les résultats possibles du test. Une instance peut alors être classée suivant l'un des chemins de la racine vers un nœud de feuille.

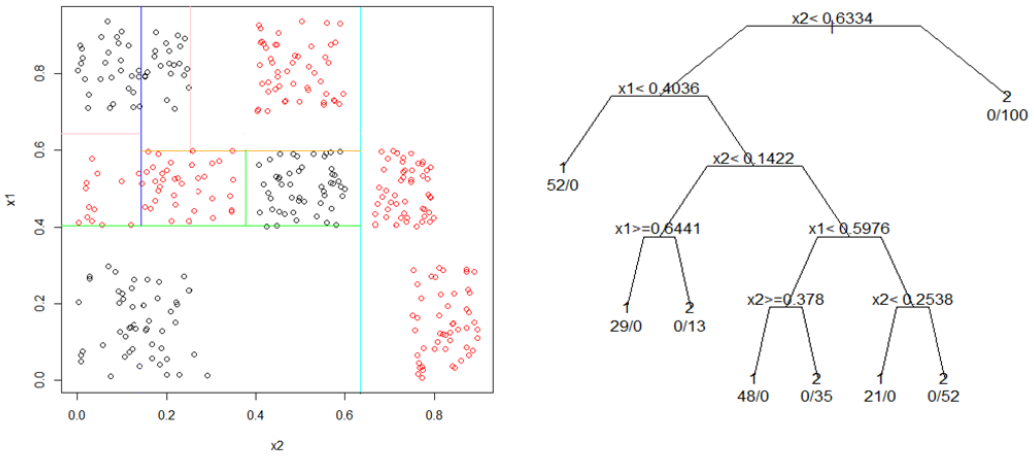


FIGURE 2.18 – Résolution d'un problème de classification binaire par le biais d'un arbre de décision. Les feuilles étant pures en termes de classes, l'arbre ne se développe plus.

Grâce à sa structure arborescente, cet algorithme a la capacité de fournir à l'utilisateur des résultats compréhensibles (sous formes de règles) qui semblent donner une structure à la variable cible apprise. Les instances obtenues suivant un certain chemin ont normalement la même classe et partagent ainsi les mêmes caractéristiques. Cet algorithme semble donc approprié pour atteindre l'objectif de la première catégorie des approches prédictives et descriptives d'une manière simultanée. Cependant, les arbres de décision sont dans certains cas incapables d'atteindre l'objectif de la deuxième catégorie des approches prédictives et descriptives d'une manière simultanée. En effet, selon la distribution des données dans l'espace d'entrée, l'arbre de décision crée naturellement des polytopes (fermés et ouverts) à l'aide des règles. La présence des polytopes ouverts empêche l'algorithme de découvrir la structure *complète* du concept cible Y. La figure 2.18 présente un exemple illustratif du fonctionnement de l'arbre de décision sur un jeu de données caractérisé par la présence de deux classes ('rouge' et 'noire'), 350 instances et deux variables descriptives $x1$ et $x2$. A partir de ce résultat, on constate que cet algorithme fusionne les deux sous-groupes de classe rouge (situés à la droite de la figure), bien que les exemples du premier sous-groupe ont des caractéristiques différentes de celles du deuxième sous-groupe. Dans le cas extrême, ces deux sous-groupes peuvent même être très éloignés et donc être de caractéristiques assez différentes.

Dans la littérature, les améliorations apportées sur la performance prédictive du modèle sont effectuées en ignorant la contrainte d'interprétation. A titre d'exemple, la présence des méthodes d'ensembles (*e.g.,* Boosting, les forêts d'arbres aléatoires, *etc*).

B.3. Le clustering supervisé

Dans la littérature, plusieurs algorithmes de clustering standard ont été soumis à des modification afin qu'ils soient adaptés au problème supervisé. Ces algorithmes sont connus sous le nom de *clustering supervisé* (ou en anglais *supervised clustering*). La différence entre le clustering standard (non supervisé) et le clustering supervisé est donnée dans la figure 2.19. Les algorithmes de clustering supervisé visent à former des clusters purs en termes de classe tout en minimisant le nombre de clusters K (la première catégorie du clustering prédictif). Cette contrainte sur K va empêcher ces algorithmes de clustering supervisé de découvrir la structure complète du concept cible. De ce fait, un seul cluster peut donc contenir un certain nombre de sous-groupes distincts (voir le cluster G de la figure 2.19 b)).

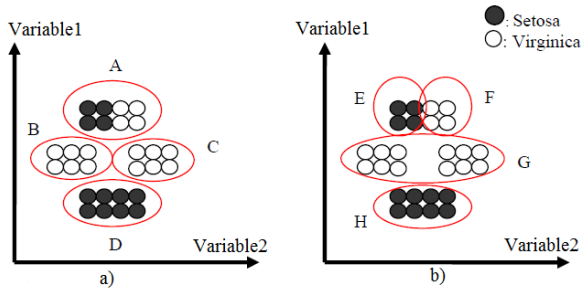


FIGURE 2.19 – La différence entre le clustering standard a) et le clustering supervisé b)

Les algorithmes de clustering supervisé les plus répandus dans la littérature sont :

• Al-Harbi *et al.* [6] proposent des modifications au niveau de l'algorithme des K-moyennes. Ils remplacent la distance euclidienne usuelle par une distance euclidienne pondérée. Le vecteur de poids est choisi de telle sorte que la confiance des partitions générées par l'algorithme des K-moyennes soit maximisée. Cette confiance est définie comme étant le pourcentage d'objets classés correctement par rapport au nombre total d'objets dans le jeu de données. Dans cet algorithme, le nombre de clusters est une entrée.

• Aguilar *et al.* [4] et Slonim *et al.* [95] ont proposé des méthodes basées sur l'approche agglomérative ascendante. Dans [4], les auteurs ont proposé un nouvel algorithme de clustering hiérarchique (S-NN) basé sur les techniques du plus proche voisin. Cet algorithme commence par N clusters où N est le nombre d'objets du jeu de données. Ensuite, il fusionne successivement les clusters ayant des voisins identiques (*i.e.,* objets proches ayant la même étiquette). Par conséquent, tous les voisins ayant les distances plus courtes que le premier ennemi (*i.e.,* l'objet qui n'a pas la même étiquette) seront collectés. Tishby *et al.* ont introduit dans [99] la méthode 'information bottleneck'. Basée sur cette méthode, ils ont proposé une nouvelle méthode de clustering (agglomérative) [95] qui maximise d'une manière explicite, l'information mutuelle entre les données et la variable cible par cluster.

• Dans [32], Cevikalp *et al.* ont proposé une méthode, nommée HC, qui crée des clusters homogènes. Ces travaux sont effectués dans le but de trouver le nombre et l'emplacement initial des couches cachées pour un réseau RBF. Cevikalp *et al.* supposent que les classes sont séparables puisqu'ils cherchent des clusters purs en classe. Le nombre et l'emplacement des clusters sont déterminés en fonction de la répartition des clusters ayant des chevauchements entre les classes. L'idée centrale de l'algorithme HC est de partir d'un nombre de clusters égal au nombre de classes

puis de diviser les clusters qui se chevauchent en tenant compte de l'information supplémentaire donnée par la variable cible.

• Eick *et al.* [46] proposent quatre algorithmes de clustering supervisés, basés sur des exemples représentatifs. Ce genre d'algorithme a pour but de trouver un sous-ensemble de représentants dans l'ensemble d'entrées de telle sorte que le clustering généré en utilisant ce dernier minimise une certaine fonction de pertinence. Dans [46], les auteurs utilisent une nouvelle fonction pour mesurer la qualité de ces algorithmes. Cette fonction remplit les deux critères suivants : *i*) minimisation de l'impureté de classe dans chaque cluster *ii*) minimisation du nombre de clusters ?

SPAM, le premier algorithme proposé par Eick et al, est une variation de l'algorithme de clustering PAM (Partitioning Around Medoids). Le deuxième algorithme proposé par les auteurs dans [46] est SRIDHCR (Single Representative Insertion/Deletion Steepest Decent Hill Climbing with Randomized Restart). Cet algorithme est un algorithme itératif. Il commence par initialiser aléatoirement un certain nombre d'exemples représentatifs. Les clusters sont alors créés en attribuant les exemples au cluster ayant le représentant le plus proche. Par la suite, l'algorithme vise à améliorer la qualité du clustering par l'ajout ou la suppression d'un exemple de l'ensemble des représentants. L'algorithme s'arrête lorsqu'aucune amélioration au niveau de la fonction de pertinence ne peut être réalisée. Le troisième algorithme proposé par ces auteurs est TDS (Top Down Splitting). Cet algorithme suit une approche descendante. Il commence par un seul cluster (*i.e.,* le cluster racine qui contient tous les exemples). Ensuite, il divise d'une manière récursive les clusters (si cette division n'entraîne pas une augmentation de la valeur de la fonction de pertinence) en remplaçant le médoïde du cluster par deux médoïdes : Le premier (respectivement le deuxième) médoïde correspond au medoid de la classe la plus fréquente (respectivement la deuxième classe fréquente) dans le cluster. Le dernier algorithme proposé par Eick et al. dans [46] est l'algorithme SCEC (Supervised Clustering using Evolutionary Computing). Cet algorithme utilise les techniques évolutionnistes pour trouver l'ensemble optimal des représentants .

Au-delà du fait que les algorithmes de clustering supervisé ne permettent pas une découverte complète du concept cible, chacun de ces algorithmes a des points faibles. A titre d'exemple :

— Les quatre algorithmes proposés par Eick *et al.* [46] sont basés sur l'optimisation d'une fonction de pertinence qui nécessite un paramètre de régularisation β. L'algorithme SPAM est une variation de l'algorithme PAM. Cet algorithme est coûteux en temps de calcul. Sa complexité est en $O(K \times (N-K)^2 \times t)$ avec N est le nombre d'instances, K est le nombre de clusters et t est le nombre d'itérations. De plus, le nombre de clusters est un paramètre utilisateur. L'algorithme SRIDHCR est aussi un algorithme coûteux en temps de calcul. Dans chaque itération, pour décider d'ajouter ou de retirer un exemple de l'ensemble des représentants, N partitions doivent être construites et évaluées. De plus, le nombre de clusters et la qualité de la partition générée par cet algorithme dépendent de l'ensemble de représentants choisi au départ. Pour choisir la meilleure partition (celle qui minimise la fonction de pertinence), SRIDHCR est exécuté r fois. Le paramètre r est à définir par l'utilisateur.
— L'algorithme de Cevikalp traite en particulier les classes ayant des chevauchements. Si une classe n'a pas de chevauchement avec les autres classes, celle-ci sera considérée comme un seul cluster bien qu'elle contienne une structure sous-jacente. De plus, cette méthode est très sensible à la présence de bruit.
— L'algorithme S-NN est basé sur l'approche agglomérative ascendante qui est une approche coûteuse en temps de calcul.
— L'algorithme de Al-Harbi *et al.* est un K-moyennes modifié. Généralement, l'algorithme des K-moyennes est caractérisé par sa complexité linéaire. Pour l'optimisation des poids,

l'algorithme de Al-Harbi *et al.* utilise un algorithme génétique. L'utilisation de sa méta-heuristique augmente le coût du modèle. En ce qui concerne le nombre de clusters, l'utilisateur doit le définir *a priori*.

B.4. Clustering prédictif basé sur les arbres (approche de Dzeroski)

Le clustering prédictif basé sur les arbres (ou en anglais predictif clustering trees "PCT") proposé par Dzeroski *et al.* dans [43], peut être présenté comme une généralisation des arbres de décision. Il peut être utilisé pour une variété de tâches d'apprentissage, y compris la prédiction et la description. Le PCT considère un arbre de décision comme une hiérarchie de clusters : la racine d'un PCT correspond à un cluster contenant l'ensemble des données qui est récursivement partitionné en des petits sous-groupes tout en se déplaçant vers le bas de l'arbre. Les feuilles représentent les clusters au niveau le plus bas de la hiérarchie et chaque feuille est étiquetée avec le prototype du cluster correspondant. L'heuristique (h) qui est utilisé pour sélectionner les tests (t) est la réduction de la variance causée par le partitionnement (P) des instances. En maximisant la réduction de la variance, l'homogénéité du cluster est maximisée et la performance prédictive est ainsi améliorée.

La principale différence entre l'algorithme de PCT et d'autres algorithmes d'apprentissage basés sur les arbres de décision est que celui-ci considère la fonction de variance et la fonction prototype (qui calcule une étiquette pour chaque feuille) en tant que paramètres qui peuvent être instanciés pour une tâche d'apprentissage donnée. L'algorithme du clustering prédictif selon Dzeroski *et al.* est présenté dans l'algorithme 1.

Entrée :
- Un ensemble d'exemple E, où chaque exemple prend la forme suivante : $O = (A, Y)$ (A est un vecteur contenant d variables descriptives et Y est une classe cible)
- Un biais de langage B qui permet de décrire les données.
- Une distance *dist* permettant de mesurer la proximité entre deux exemples donnés.
- Une fonction p, dite prototype, permettant d'affecter à chaque exemple une étiquette.

Sortie :
- Chaque cluster est associé avec une description exprimée par le biais de langage B.
- Chaque cluster a une prédiction exprimée par le prototype.
- Inertie intra cluster est minimale (similarité maximale).
- Inertie inter-clusters est maximale (similarité minimale).

Algorithme 1 – Le clustering prédictif selon Dzeroski *et al.*

B.5. Le clusterwise

L'objectif de la régression linéaire typologique (ou clusterwise) est de déterminer une partition d'un ensemble de N instances en K clusters obtenus selon un modèle de régression linéaire reliant une variable y à un ensemble de variables explicatives $\{x_j, j = 1, \ldots, d\}$. On note X la matrice des données associée aux variables explicatives. Cela revient à supposer l'existence d'une variable latente qualitative C à K modalités telle que $E(y|x) = b_0^k + b_1^k x_1 + b_2^k x_2 + \ldots + b_d^k x_d$ où les b_j^k sont les coefficients de la régression de y sur les x_j restreinte aux N_k observations de la classe k.

décrites par y^k, X^k ; avec $N_k \geq d$ pour garantir l'existence d'une solution pour les b^k. La régression typologique (ou clusterwise) revient donc à chercher simultanément une partition en K clusters et et le vecteur b^k des coefficients b_j^k correspondant minimisant le critère $Z = \sum_{k=1}^{K} \parallel X^k b^k - y^k \parallel^2$

Diverses méthodes et algorithmes ont été proposés pour l'estimation de ces coefficients. On peut par exemple citer les travaux de DeSarbo et Cron [42] qui utilisent une méthode du maximum de vraisemblance et l'algorithme EM pour estimer les paramètres du modèle.

La régression linéaire typologique (ou clusterwise) fait l'objet de nombreuses publications, en association avec des données fonctionnelles (Preda et Saporta, [86]), des données symboliques (de Carvalho et al., [39]), dans des cas multiblocs (De Roover et al., [40]).

2.6 Conclusion : notre objectif

2.6.1 Objectif

Dans cette thèse, nous nous sommes fixés comme objectif de développer un algorithme d'apprentissage "interprétable" qui permet de décrire et de prédire d'une manière simultanée. Pour ce faire, nous proposons d'adapter un algorithme de clustering au problème de la classification supervisée. Autrement dit, l'idée est de modifier un algorithme de clustering afin qu'il soit un bon prédicteur tout en gardant sa faculté à bien décrire les données et donc le concept cible à apprendre. Cet algorithme doit également fournir des résultats facilement interprétables par l'utilisateur. Ce type d'algorithme est connu sous le nom de clustering prédictif.

Le modèle d'apprentissage recherché au cours cette thèse est un modèle qui traite principalement trois différents axes, à savoir, la description, l'interprétation et la prédiction (voir la figure 2.20). D'une part, l'utilisation d'un algorithme de clustering donne une garantie immédiate sur l'axe de description. Il s'agit de décrire l'ensemble des données à l'aide de la découverte de la structure sous-jacente existante dans celui-ci. Cependant, le concept de similarité utilisé dans le cadre du clustering traditionnel ne prend pas en considération l'appartenance des instances à des classes différentes. Par conséquent, deux instances similaires de classes différentes peuvent être fusionnées ensemble. Ceci produit une détérioration au niveau de la performance prédictive du modèle en question. D'où la nécessité d'incorporer l'information donnée par la variable cible dans le processus du clustering afin d'assurer l'axe de prédiction et donc la découverte de la structure interne de la variable cible. D'autre part, les algorithmes de partitionnement tels que les K-moyennes fournissent en général des partitions dont chaque groupe est représenté par un prototype. L'utilisation dans ce cas d'un biais de langage est nécessaire pour rendre les résultats issus de cet algorithmes plus interprétables par l'utilisateur.

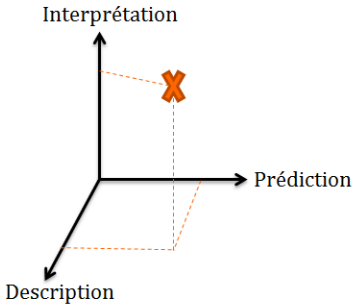


FIGURE 2.20 – Les trois axes traités dans cette thèse

D'une manière générale, le modèle du clustering prédictif recherché au cours de cette thèse doit prendre en considération les points suivants :

1. **La maximisation de la performance prédictive du modèle** : Comme dans le cadre de la classification supervisée, le but majeur des algorithmes de clustering prédictif est de prédire correctement la classe des nouvelles instances.
2. **La découverte de la structure interne de la variable cible** : Dans la phase d'apprentissage, ce modèle doit être capable de découvrir la structure sous-jacente de l'ensemble des données tout en tenant compte de l'étiquetage des instances.
3. **La facilité de l'interprétation des résultats** : Dans les domaines critiques (e.g., le service marketing à Orange), l'interprétation des résultats générés par un système d'apprentissage est une question incontournable. Pour cette raison, on souhaite avoir un modèle interprétable même par des utilisateurs non experts.
4. **La minimisation des connaissances,** *a priori* **requises, de la part de l'utilisateur (***i.e.,* **pas ou peu de paramètres utilisateur)** : Le modèle recherché doit être un algorithme qui contient très peu ou aucun paramètres utilisateur.
5. **La minimisation de la taille (complexité) du modèle prédictif - descriptif** : Le modèle recherché doit être efficace, rapide en termes de calcul et facile à implémenter.

En tenant compte de tous les points cités ci-dessus et en adoptant la définition donnée par Dzeroski *et al.* dans [43], le clustering prédictif peut être présenté par l'algorithme 2.

Entrée :

- Un ensemble de données $\mathcal{D}$, où chaque instance X_i est décrite par un vecteur de d dimensions et par une classe $Y_i \in \{1, \ldots, J\}$.
- Un ensemble de prototypes initiaux (ou centres initiaux) qui forment la partition initiale.
- Une distance *dist* qui mesure la proximité entre deux instances.
- Un biais de langage B permettant de décrire les données.

Sortie :

- Chaque cluster est représenté par un prototype qui possède la même étiquette de classe.
- Chaque cluster est associé avec une description donnée par le biais de langage B.

Si (*l'algorithme est dédié à la première catégorie*) **Alors**
- Le nombre de clusters (K) est minimal.
- Le taux de bonnes classifications est maximal.

Fin Si

Si (*l'algorithme est dédié à la deuxième catégorie*) **Alors**
- L'inertie intra-clusters est minimale (l'homogénéité des instances est maximale).
- L'inertie inter-clusters est maximale (la similarité entre les clusters est minimale).
- Le taux de bonnes classifications est maximal.

Fin Si

Algorithme 2 – Algorithme générique du clustering prédictif

Les approches issues de la littérature qui semblent adéquates pour résoudre notre problématique sont les approches cités dans la section 2.5.2. Cependant, ces algorithmes ne prennent pas en considération tous les points cités ci-dessus. Dans cette thèse, nous avons choisi de modifier l'algorithme de clustering le plus répandu dans la littérature, à savoir l'algorithme des K-moyennes. La version modifiée de celui-ci sera nommée *les K-moyennes prédictives*.

2.6.2 K-moyennes prédictives

La méthode des centres mobiles (ou les K-moyennes) due à Forgy [49] permet de subdiviser l'ensemble des individus en un certain nombre de classes en employant une stratégie d'optimisation itérative dont le principe général est de générer une partition initiale, puis de chercher à l'améliorer en réattribuant les données d'une classe à l'autre. Cet algorithme recherche des maxima locaux en optimisant une fonction objectif traduisant le fait que les individus doivent être similaires au sein d'une même classe, et dissimilaires d'une classe à une autre. Les classes de la partition finale, prises deux à deux, sont d'intersection vide et chacune est représentée par un noyau (ou prototype).

L'algorithme des *K-moyennes prédictives* consiste à prédire la classe d'une nouvelle instance en se basant sur sa proximité à un des groupes formés dans la phase d'apprentissage. Plus précisément, dans le problème du clustering prédictif, les données d'apprentissage $\mathcal{D} = \{X_i\}_{i=1}^{N}$ sont composées de N instances. Chaque instance i est décrite par d variables descriptives $\{X_i^1, \ldots, X_i^d\}$ et une variable cible Y_i contenant l'information de la classe. On notera $Y_i \in \{1, \ldots, J\}$ où J est le nombre de classes. L'objectif de l'algorithme des K-moyennes prédictives est donc de former, à partir des données d'apprentissage, K clusters purs et homogènes : les instances appartenant à un cluster $k \in \{1, \ldots, K\}$ doivent, d'une part, avoir la même classe j et d'autre part être différentes des instances appartenant à d'autres clusters. A la fin du processus d'apprentissage, une technique est utilisée pour étiqueter chaque cluster appris (*e.g.,* l'utilisation du vote majoritaire). Au final, la prédiction d'une nouvelle instance se fait selon son appartenance à un des clusters appris. Autrement dit, cette nouvelle instance reçoit j comme prédiction si elle est plus proche du centre de gravité du cluster de classe j (*i.e.,* utilisation du 1 plus proche voisin). Les différentes étapes de l'algorithme des K-moyennes prédictives sont présentées dans l'algorithme 3.

Étape 1 : Prétraitement des données
Étape 2 : Initialisation des centres
Étape 3 : Répéter un certain nombre de fois (R) jusqu'à convergence
  3.1 Cœur de l'algorithme
Étape 4 : Choix de la meilleure convergence
Étape 5 : Mesure d'importance des variables (après la convergence et sans réapprendre le modèle)
Étape 6 : Affectation des classes aux clusters appris.
Étape 7 : Prédiction de la classe des nouveaux exemples.

Algorithme 3 – Les étapes des K-moyennes prédictives.

Pour aboutir à notre objectif, chaque étape de l'algorithme des K-moyennes (Algorithme 3) pourrait être traitée individuellement. L'idée est de tester à quel point la supervision de chaque

étape pourrait aider l'algorithme des K-moyennes standard à remplir la tâche du clustering prédictif. Au final, on pourrait obtenir un algorithme des K-moyennes supervisé à chaque étape. Cet algorithme sera comparé par la suite aux approches potentielles décrites dans la section 2.5.2 B. Dans cette thèse on ne s'intéresse qu'à la modification de quatre étapes de l'algorithme des K-moyennes standard. Ces étapes sont présentées dans ce qui suit :

L'étape 1 des K-moyennes prédictives (voir Algorithme 3) fait l'objet du **chapitre 3** de ce mémoire. L'étape du prétraitement des données est une étape primordiale que ce soit dans le cadre du clustering classique ou dans le cadre du clustering prédictif. L'intérêt de cette étape est d'incorporer l'information donnée par la variable cible dans les données dans le but de permettre à l'algorithme des K-moyennes standard de la prendre en considération. En effet, lorsqu'on dispose d'un ensemble de données où les instances de différentes classes sont proches les unes des autres en termes de distance, l'application de l'algorithme classique des K-moyennes sur ces données va entrainer une détérioration au niveau de la performance prédictive (ces instances proches vont être fusionnées ensemble indifféremment à leur classe d'appartenance). Le but de ce chapitre est donc de définir une distance dépendante de la classe cible qui vérifie que deux instances proches en termes de distances sont également proches en termes de leur comportement vis-à-vis de la variable cible. Cette distance peut être écrite à l'aide d'un prétraitement supervisé des données. Pour respecter les points imposés dans la section 2.6.1, le prétraitement proposé doit impérativement être interprétable (point 3.), robuste (point 1.), rapide (point 5.), et sans paramètres utilisateur (point 4.). Avec ce genre de prétraitement de données on espère augmenter la performance prédictive de l'algorithme classique des K-moyennes comparant à sa performance prédictive en utilisant des prétraitements non supervisés.

L'étape 2 des K-moyennes prédictives (voir Algorithme 3) fait l'objet du **chapitre 4** de ce mémoire. L'un des inconvénients de l'algorithme des K-moyennes standard réside dans sa sensibilité envers le choix des centres initiaux. En effet, l'étape d'initialisation influence la qualité de la solution trouvée ainsi que le temps d'exécution [29]. Lors de déséquilibre des classes à prédire (par exemple, l'existence d'une classe majoritaire et d'une classe minoritaire) dans l'ensemble des données, l'utilisation d'une méthode d'initialisation non supervisé s'avère insuffisante. En effet, la probabilité de choisir plus d'un centre dans la classe majoritaire et de ne choisir aucun centre dans la classe minoritaire est très grande. Par conséquent, une détérioration au niveau de la performance prédictive du modèle peut être produite. A partir de ce constat, il est naturel de se demander si l'utilisation d'une méthode d'initialisation supervisée pourrait aider l'algorithme des K-moyennes standard à remplir la tâche du clustering prédictif. Plus précisément, cette étape traite principalement le premier point cité dans la section 2.6.1 qui est la maximisation de la performance prédictive du modèle.

L'étape 4 des K-moyennes prédictives (voir Algorithme 3) fait l'objet du **Chapitre 5** de ce mémoire. L'algorithme des K-moyennes n'assure pas de trouver un minimum global. Il est souvent exécuté plusieurs fois (on parle de "réplicates") et la meilleure solution en termes d'erreur quadratique moyennes est alors choisie. Dans le cadre du clustering prédictif, la notion de "meilleure solution" diffère de celle connue dans le cadre du clustering standard. Pour la première catégorie du clustering prédictif où on l'impose d'avoir un nombre faible de clusters (voir la figure 2.14 de la section 2.5.2), l'axe privilégié dans ce cas est l'axe de prédiction. Un critère supervisé tel que l'indice de rand ajusté peut être utilisé pour mesurer la qualité des résultats et donc choisir la meilleure partition. Pour la deuxième catégorie du clustering prédictif où l'on cherche à découvrir la structure complète de la variable cible, l'utilisation des critères proposés

dans la littérature s'avère insuffisant. En effet, dans ce cadre d'étude, on cherche à réaliser le bon compromis entre la prédiction et la description et à notre connaissance, il n'existe pas dans la littérature un critère analytique permettant d'évaluer ce compromis. Le but de ce chapitre est donc de proposer un critère analytique permettant de mesurer la qualité des résultats générés par la deuxième catégorie du clustering prédictif. Cette étape cherche à traiter les deux premiers points cités dans la section 2.6.1, à savoir, la maximisation de la performance prédictive du modèle et la découverte de la structure interne de la variable cible.

L'étape 5 des K-moyennes prédictives (voir Algorithme 3) fait l'objet de l'annexe E de ce mémoire. Pour une interprétation aisée des résultats générés par l'algorithme des K-moyennes prédictives, on cherchera dans cette partie de la thèse à proposer une méthode supervisée permettant de mesurer l'importance des variables selon leurs contributions dans le processus d'apprentissage. Partant d'une partition de référence (partition à interpréter), l'importance de chaque variable sera définie alors comme le pouvoir prédictif de celle-ci à bien prédire cette partition de référence. Ce pouvoir prédictif est mesuré à l'aide de un arbre de décision. Cette étape traite principalement le troisième point cité dans la section 2.6.1 à savoir, la facilité de l'interprétation des résultats. Puisque le travail effectué au cours de cette thèse sur ce sujet n'est pas encore achevé, nous avons choisi donc de le placer dans un annexe au lieu de le considéré comme un chapitre.

Le chapitre 6 présente d'une part une synthèse des résultats obtenus dans les chapitres précédents et d'autre part, il présente une comparaison de l'algorithme des K-moyennes prédictives proposé dans cette thèse (intégrant les différentes étapes supervisées) avec d'autres méthodes issus de la littérature. Cette partie expérimentale est divisée en deux grandes parties. La première partie se focalise sur le côté prédictif du modèle. Tandis que la deuxième partie se focalise sur l'aspect interprétable du modèle et la capacité de celui-ci à bien découvrir la structure interne de la variable cible. Pour finir, une conclusion dresse le bilan des trois années de thèse, des travaux réalisés et des travaux futurs. Nous rappelons les différentes notions introduites dans la thèse, ainsi que les résultats obtenus.

Chapitre 3

Distance dépendante de la classe

Sommaire

Ce chapitre a fait l'objet de la publication suivante :

[10] Oumaima Alaoui Ismaili, Vincent Lemaire, Antoine Cornuéjols. Supervised pre-processings are useful for supervised clustering. In *Springer Series Studies in Classification, Data Analysis, and Knowledge Organization*, Bremen, 2015.

3.1 Introduction

La question évoquée dans cette thèse "*comment décrire et prédire d'une manière simultanée ?*" conduit à un nouvel aspect de l'apprentissage. Ce dernier est appelé *"le clustering prédictif"* (voir Algorithme 2 du Chapitre 2). Les algorithmes appartenant à ce type d'apprentissage peuvent être catégorisés en fonction des besoins de l'utilisateur. La première catégorie regroupe l'ensemble des algorithmes de clustering "modifiés" permettant de prédire correctement la classe des nouvelles instances sous contrainte d'avoir un nombre minimal de clusters dans la phase d'apprentissage (voir la partie gauche de la figure 3.1). Cette catégorie met l'accent sur l'axe de prédiction tout en "ignorant" l'axe de description. La deuxième catégorie regroupe l'ensemble des algorithmes permettant tout d'abord de découvrir la structure interne *complète* de la variable cible, puis munie de cette structure prédire correctement la classe des nouvelles instances (voir la partie droite de la figure 3.1). Contrairement aux algorithmes de la première catégorie, ces algorithmes cherchent à réaliser un compromis entre la description et la prédiction sans privilégier un axe par rapport à l'autre.

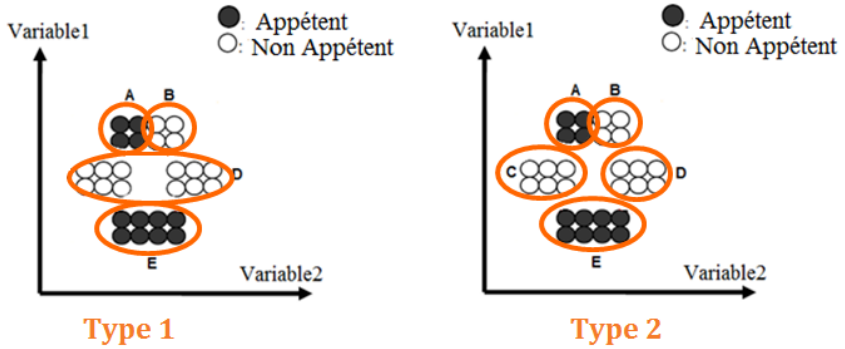


FIGURE 3.1 – Les deux types des algorithmes de clustering prédictif

Les algorithmes classiques de clustering visent à subdiviser l'ensemble des instances en un certain nombre de groupes (ou clusters) de telle sorte que les instances au sein de chaque cluster doivent être similaires entre elles et dissimilaires des instances des autres clusters. Pour l'algorithme des K-moyennes, cette notion de similarité/dissimilarité est représentée par une distance ou une métrique. Deux instances sont considérées comme similaires si et seulement si elles sont proches en termes de distance (la distance la plus utilisée est la distance Euclidienne). En revanche, dans le cadre des K-moyennes prédictives, la similarité entre deux instances n'est pas uniquement liée à leur proximité en termes de distance mais elle est également liée à leur ressemblance en termes de leur classe d'appartenance : deux instances sont similaires si et seulement si elles sont proches en termes de distance **et** appartiennent à la même classe. De ce fait, l'utilisation d'une distance ne prenant pas en compte l'information de la classe reste insuffisante : deux instances proches en termes de distance vont être considérées comme similaires indifféremment à leur classe d'appartenance et donc la probabilité que l'algorithme les regroupe ensemble sera élevée. À titre d'exemple, les instances appartenant aux deux groupes A et B de la figure 3.1.

Pour permettre à l'algorithme de K-moyennes classique de prendre en considération l'information donnée par la variable cible et ainsi améliorer sa performance prédictive, deux voies peuvent être exploitées. La première voie consiste à modifier la fonction du coût de l'algorithme des K-moyennes afin de proposer une nouvelle fonction objectif capable d'établir une certaine relation entre la similarité classique pour les instances et leur classe d'appartenance. À titre d'exemple, Peralta et al. ont défini dans [85] une nouvelle fonction objectif qui s'écrit sous forme

d'une combinaison convexe entre la fonction objectif usuelle de l'algorithme de K-moyennes et sa version supervisée. Cependant, cette fonction nécessite un paramètre utilisateur pour équilibrer les deux scores ce qui requière une phase d'ajustement (validation croisée pour trouver la bonne valeur du paramètre). La deuxième voie consiste à incorporer l'information donnée par la variable cible dans les données sans modifier la fonction du coût de l'algorithme des K-moyennes classique. Dans ce chapitre, on s'intéresse exclusivement à l'étude de la deuxième voie.

La démarche suivie pour incorporer l'information donnée par la variable cible dans les données doit impérativement respecter le point crucial qu'un algorithme des K-moyennes prédictives doit posséder, à savoir, l'interprétabilité des résultats (voir la section 2.6.1 du chapitre 2). L'intérêt de cette démarche est d'une part de rendre la tâche d'interprétation des résultats plus aisée pour l'algorithme des K-moyennes standard. D'autre part, elle permet de modifier indirectement la fonction du coût de l'algorithme des K-moyennes standard dans le but de l'aider à atteindre l'objectif des K-moyennes prédictives. Pour atteindre cet objectif, la méthode recherchée doit être capable de générer une distance, dite supervisée, permettant d'établir une certaine relation entre la similarité classique entre les instances et leur classe d'appartenance.

Dans ce chapitre, on suppose qu'une estimation des distributions uni-variées conditionnelles aux classes $P(\mathcal{X}|C)$ pourrait aider l'algorithme des K-moyennes standard à atteindre l'objectif mentionné ci-dessus. Parmi les méthodes permettant cette estimation probabiliste, on choisit de s'orienter vers les méthodes les plus interprétables, à savoir : les méthodes supervisées de discrétisation pour les variables continues et les méthodes supervisées de groupage en modalités pour les variables catégorielles. Ces méthodes cherchent à trouver la partition des valeurs ou des modalités qui donne le maximum d'informations sur la répartition des J classes connaissant les intervalles de discrétisation ou les groupes de modalités. A titre d'exemple, la figure 3.2 présente la discrétisation supervisée des deux variables continues (Variable 1 et Variable 2) pour un jeu de données caractérisé par la présence de deux classes. Pour cet exemple illustratif, la variable 1 est divisée en 3 intervalles tandis que la variable 2 est divisée en 4 intervalles.

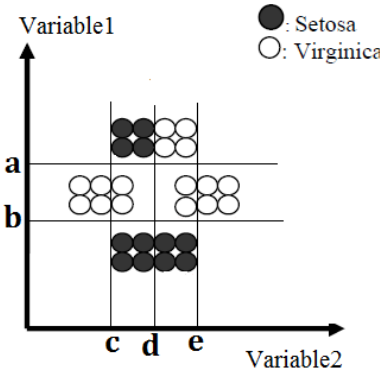


FIGURE 3.2 – Discrétisation supervisée

Une façon intuitive permettant d'exploiter les informations générées par les méthodes de discrétisation et de groupage en modalités est de considérer pour chaque instance l'appartenance de ses valeurs à un intervalle ou à un groupe de modalités (voir, Figure 3.3). Suivant cette démarche, chaque instances X_i sera transformée en $\sum_{l=1}^{d} t_l$ (t_l est le nombre d'intervalles ou groupes de modalité issu de la variable l et d est le nombre des variables descriptives) variables booléennes. Dans ce cas, pour mesurer la similarité entre les instances, la distance de Hamming est utilisée. Cette distance vérifie la propriété suivante : *deux instances proches en termes de distance sont également proches en termes de leurs appartenances aux mêmes intervalles de*

discrétisation (ou aux groupes de modalités selon la nature des variables descriptives). Cette méthode est appelée par la suite **Binarization** (BIN-BIN). Pour plus de détails, voir la section 3.2.2 de ce chapitre.

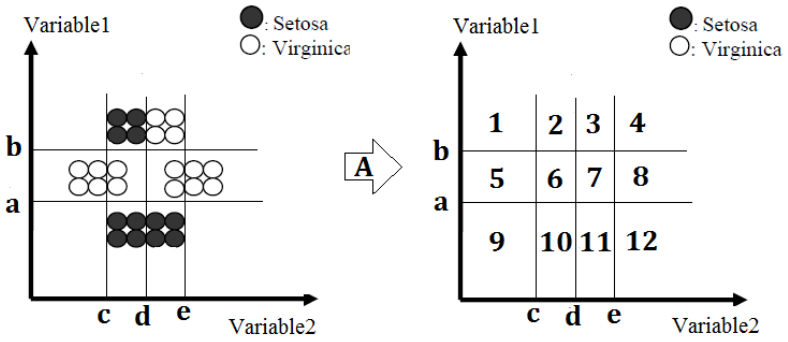


FIGURE 3.3 – L'exploitation des données à travers l'appartenance aux intervalles de discrétisation

Une autre façon permettant d'exploiter les informations générées par les méthodes de discrétisation et de groupage en modalités est de considérer la quantité d'informations contenue dans les intervalles ou dans les groupes de modalités conditionnellement aux classes "$P(X \in I|C_j)$", $j \in \{1, \ldots, J\}$" où I est le nombre d'intervalles ou de groupes de modalités pour une variable donnée (voir Figure 3.4). Suivant cette démarche, chaque instance X_i sera transformée en $J \times d$ variables numériques : chacune des d variables d'origines est transformée en J synthétiques variables $(log(P(X_i \in I|C_1)), \ldots, log(P(X_i \in I|C_J)))$. Dans ce cas, pour mesurer la similarité entre les instances, la distance bayésienne est utilisée. Cette distance vérifie la propriété suivante : *deux instances proches en termes de distance sont également proches en termes de leur probabilité d'appartenir à la même classe.* Cette méthode est appelée par la suite **Conditional Info** (CI-CI). Pour plus de détails voir la section 3.2.3 de ce chapitre.

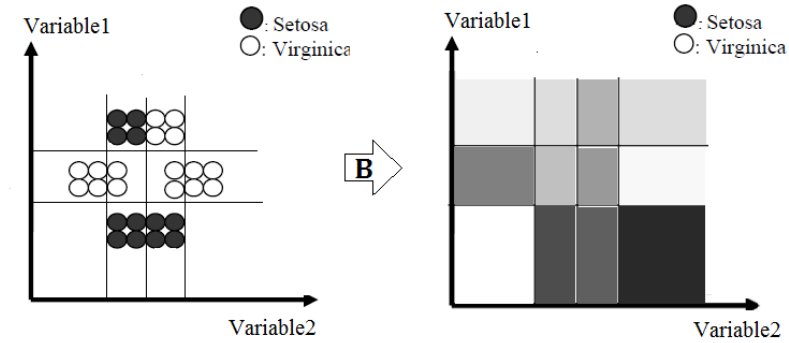


FIGURE 3.4 – L'exploitation des données à travers le calcul de la quantité d'information contenue dans les intervalles

Le reste de ce chapitre est organisé comme suit : la section 3.2.1 définit le principe de la discrétisation pour les variables continues et du groupage en modalités pour les variables catégorielles. Afin d'exploiter les informations générées par ces méthodes, les deux sections 3.2.2 et 3.2.3 présentent respectivement l'approche Binarization (BIN-BIN) et l'approche Conditional Info (CI-CI). Pour étudier l'impact de l'utilisation d'une distance supervisée (à travers l'utilisation du prétraitement supervisé BIN-BIN ou CI-CI) sur la qualité (au sens du clustering prédictif) des résultats fournis par l'algorithme des K-moyennes standard, une étude expérimen-

tale sera menée dans la section 3.4. L'objectif de cette section est de chercher à répondre à la question suivante : *les méthodes de prétraitement supervisées pourraient-elles aider l'algorithme des K-moyennes standard à atteindre l'objectif du clustering prédictif ?* Finalement, et avant de conclure dans la section 3.6, les deux axes de description et d'interprétation seront discutés dans la section 3.5.

3.2 Distance dépendante de la classe

L'incapacité de l'algorithme des K-moyennes standard à atteindre l'objectif des algorithmes du clustering prédictif s'illustre essentiellement dans le cas de la non corrélation entre les classes et les clusters. C'est le cas où au moins l'une des régions denses est caractérisée par la présence d'au moins deux classes. La figure 3.5 présente un exemple illustratif de la présence d'une région dense ($\{A \cup B\}$) contenant deux classes. En effet, l'algorithme des K-moyennes standard considère deux instances proches en termes de distance comme similaires indifféremment de leur classe d'appartenance. Dans le cadre du clustering prédictif, ceci peut générer une détérioration au niveau de la performance prédictif du modèle.

FIGURE 3.5 – Cas de la non corrélation entre les classes et les clusters

Pour surmonter ce problème, l'incorporation de l'information donnée par la variable cible dans les données s'avère nécessaire. Cette incorporation pourrait aider l'algorithme des K-moyennes standard à prendre en considération l'appartenance des instances aux classes. Dans ce chapitre, on suppose que l'estimation des distributions uni-variées conditionnelles aux classes ($P(\mathcal{X}|C)$) pourrait aider cet algorithme à atteindre l'objectif souhaité.

3.2.1 Estimation des densités conditionnelles aux classes

Comme signalé dans la section 2.6.1 du chapitre 2, les algorithmes du clustering prédictif sont des algorithmes qui fournissent des résultats facilement interprétables par l'utilisateur. De ce fait, lors de la recherche de la méthode permettant d'insérer l'information donnée par la classe cible dans les données, la contrainte d'interprétabilité doit impérativement être respectée.

Parmi les méthodes les plus "*interprétables*" permettant une estimation des densités conditionnellement aux classes, on trouve les méthodes de discrétisation supervisées pour les variables continues et le groupage en modalités pour les variables catégorielles.

Pour les variables continues, la discrétisation supervisée consiste à diviser le domaine de chaque variable en un nombre fini d'intervalles identifiés chacun par un code $I_l, l \in \{1, \ldots, t\}$. Elle vise à trouver la partition des valeurs qui donne le maximum d'informations sur la répartition des J classes connaissant l'intervalle de discrétisation $I_l, l \in \{1, \ldots, t\}$. Cette partition optimale est décrite par la table de contingence comme illustrée dans le tableau 3.1.

	I_1	I_2	...	I_t	Somme
C_1	n_{11}	n_{12}	...	n_{1t}	$n_{1.}$
C_2	n_{21}	n_{22}	...	n_{1t}	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
C_J	n_{J1}	n_{J2}	...	n_{Jt}	$n_{J.}$
Somme	$n_{.1}$	$n_{.2}$	...	$n_{.t}$	N

TABLE 3.1 – Table de contingence pour une variable continue

Dans la littérature, il existe une variété de méthodes de discrétisation ou de groupage en modalités selon la nature des variables descriptives. Les méthodes les plus répandues sont notamment MODL [23], ChiSplit [21], BalancedGain [67] et l'arbre de décision C4.5 ou CART [87]. Le choix de la méthode qu'on va utiliser sera conditionnée par certains points, à savoir : la robustesse, la rapidité, la précision et finalement la minimisation des connaissances *a priori* (*i.e.,* pas (ou très peu) de paramètres utilisateur). Parmi les méthodes de discrétisation et de groupages en modalités qui respectent l'ensemble de ces points, on trouve la méthode MODL, proposé par Boullé dans [23]. Il est important de noter que ce choix n'est pas une obligation. D'autres méthodes peuvent également être utilisées à condition qu'elles respectent les points cités ci-dessus.

L'approche MODL considère la discrétisation comme un problème de sélection de modèle. Ainsi, une discrétisation est considérée comme un modèle paramétré par le nombre d'intervalles, leurs bornes et les effectifs des classes cibles sur chaque intervalle. La famille de modèles considérée est l'ensemble des discrétisations possibles. Cette famille est dotée d'une distribution a priori hiérarchique et uniforme à chaque niveau. Pour plus de détails sur cette approche voir [23].

Après cette étape dite étape de préparation supervisée des données, chaque variable est représentée par une table de contingence. Pour pouvoir exploiter les connaissances utiles existant dans ces tables, la recherche d'une méthode de recodage s'avère nécessaire. Cette méthode doit être capable de générer une distance dépendante de la classe permettant d'établir une certaine relation entre la similarité usuelle et le comportement des instances vis-à-vis de la classe cible.

Une façon permettant une exploitation aisée des tables de contingence est de considérer l'appartenance des valeurs de chaque instance aux intervalles de discrétisation ou aux groupes de modalité selon la nature des variables descriptives. Cette approche est appelée dans ce qui suit **Binarization** (BIN-BIN).

3.2.2 Binarization (BIN-BIN) - Distance de Hamming

L'approche la plus intuitive permettant d'exploiter les informations données par les tables de contingences est la considération de l'appartenance des valeurs de chaque instance aux intervalles de discrétisation ou aux groupes de modalités (à titre d'exemple, voir la figure 3.3). Il s'agit ici de transformer chaque variable en t variables booléennes ; où t est le nombre d'intervalles (ou groupes de modalités) généré par la méthode de discrétisation (ou de groupage en modalités). Cette opération de transformation est basée sur un recodage disjonctif complet : la variable synthétique prend 1 comme valeur si la valeur de la variable d'origine appartient à l'intervalle en question et elle prend zéro comme valeur dans le cas contraire. À titre d'exemple, la variable 1 du jeu de données présenté dans la figure 3.6 est transformée en 3 (nombre d'intervalles : $]-\infty, a]$,$]a, b]$ et $]b, +\infty]$) variables booléennes ($\{1, 0, 0\}$, $\{0, 1, 0\}$ et $\{0, 0, 1\}$) avec $\{1, 0, 0\}$ signifie que la valeur de la variable d'origine appartient au premier intervalle. Suivant ce processus, chaque instance

X_i sera transformée en un vecteur booléen de dimension $\sum_{l=1}^{d} t_l$. Cette approche est nommée **Binarization** (BIN-BIN).

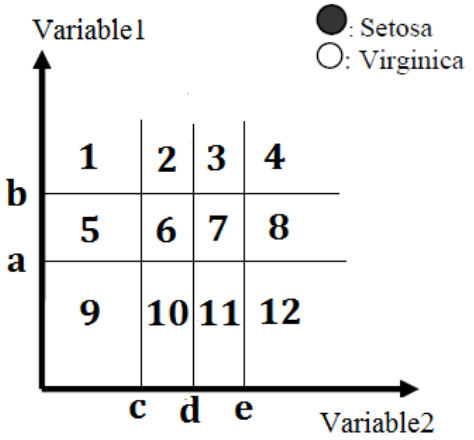


FIGURE 3.6 – Exemple illustratif d'appartenance aux intervalles après une opération discrétisation

Étant donné que les nouvelles variables synthétiques après l'opération de recodage sont des variables booléennes, une distance entre les individus appropriée à utiliser à ce stade est la distance de Hamming. On définit la distance de Hamming sur les attributs recodés comme le nombre d'attributs recodés de façon différentes. La formule mathématique de cette distance pour deux instances X_1 et X_2 de dimension p est donnée par l'équation 3.1.

$$\forall X_{1j}, X_{2j} \in \mathcal{D}, (j \in \{1, \ldots, d\}) \quad dist_H(X_{1j}, X_{2j}) = \#\{j : X_{1j} \neq X_{2j}\} \tag{3.1}$$

À partir de ce recodage, on constate que deux instances ayant une distance d_H nulle seront alors associées à la même prédiction de classe cible. À titre d'exemple, les instances appartenant au block 5 de la figure 3.6 sont quasiment de la même classe en raison de l'utilisation d'une méthode de discrétisation supervisée et elles sont présentées sous forme d'un vecteur de dimension $7 = 3+4$ prenant la forme suivante : $\{0, 1, 0, 1, 0, 0, 0\}$. De ce fait, la conclusion qui peut être tirée est que les distances faibles sont corrélées avec des instances ayant des comportements similaires vis-à-vis de la classe cible.

Avantages

— La méthode Binarization est une méthode capable de distinguer les instances selon leurs appartenances aux intervalles de discrétisation ou aux groupes de modalités de telle sorte que deux instances proches en termes de distance vont être proches en termes de leurs appartenances aux intervalles. On remarque que plus la distance est petite plus le comportement des instances vis-à-vis de la classe sera proche.
— La distance Hamming utilisée pour la méthode Binarization a l'avantage de se calculer simplement sous forme d'une distance L1 suite à un recodage binaire disjonctif complet sur chacun des attributs.
— Elle a également l'avantage d'une normalisation par rapport à la distance euclidienne.

Limites

— La qualité de la corrélation entre la distance de Hamming et le comportement en prédiction est difficile à évaluer.
— La distance de Hamming est peu discriminante : deux recodages différents peuvent correspondre à des comportements très ou peu différents vis-à-vis de la classe cible. À titre d'exemple, la figure 3.7 présente une discrétisation en 6 intervalles de la 7^{eme} variable "V7" de la base de données Waveform [2] pour un problème à trois classes (0, 1 et 2). L'axe des abscisses de la figure 3.7 représente l'ensemble des valeurs que peut prendre la variable V7. L'axe des ordonnées quant à lui, représente l'ensemble des probabilités d'appartenir à une des classes conditionnellement aux intervalles de discrétisation conditionnellement (par exemple, la courbe orange correspond à la classe 2). Les barres verticales désignent les intervalles de discrétisation (au nombre de 6). Clairement, on souhaiterait que deux instances recodées sur le $3ième$ et le $4ième$ intervalle (mélanges presque homogènes des trois classes) soient considérées comme plus proches que deux instances recodées sur le 1er intervalle (classe 2 largement majoritaire) et le $6ième$ intervalle (classe 2 absente) ce qui n'est pas le cas lorsque le recodage Binarization est utilisé.

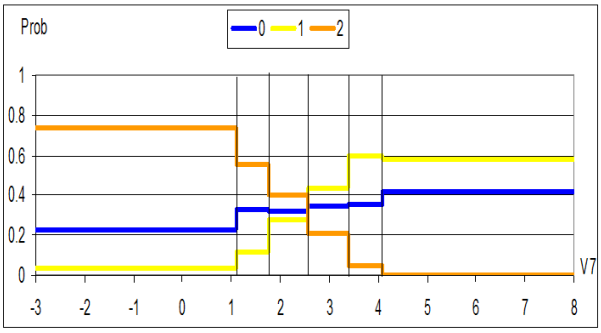


FIGURE 3.7 – Discrétisation de la variable V7 de la base Waveform via l'approche MODL

Pour surmonter les difficultés rencontrées par l'approche Binarization, la section 3.2.3 présente une nouvelle approche de recodage qui prend en considération la quantité d'information existant dans les intervalles de discrétisation (ou groupes de modalités) conditionnellement aux classes. Cette approche est appelée **Conditional Info** (CI-CI).

3.2.3 Conditional Info (CI-CI) - Distance bayésienne

Après la préparation supervisée des données, une seconde étape est mise en place dans le but d'exploiter les informations données par la discrétisation et le groupage en modalités. Cette étape est une étape de recodage où chaque variable de X_i est recodée en une variable qualitative contenant I_J valeurs de recodage. Chaque instance X_i ($i \in \{1, \ldots, N\}$) des données est alors recodée sous forme d'un vecteur de modalités discrètes $\hat{X}_i = X_{i1_1}, \ldots, X_{i1_J}, \ldots, X_{id_1}, \ldots, X_{id_J}$ où X_{id_J} représente la valeur du recodage de la variable d pour la classe J ($X_{id_J} = log(P(X_{id}|C_J))$). Ainsi, les variables de départ sont alors toutes représentées sous une forme numérique, sur un vecteur de $d \times J$ composantes : $log(P(X_{im}|C_J)), i \in \{1, \ldots, N\}, m \in \{1, \ldots, d\}$. Les deux étapes (discrétisation supervisée et recodage) forment une méthode de prétraitement supervisé des données que l'on appelle '*Conditional Info*' (CI-CI).

2. La base de données Waveform est une base de données de l'UCI caractérisée par la présence de 21 variables descriptives et une variable cible contenant 3 classes (0, 1 et 2).

Par exemple, pour un problème de classification binaire (*i.e.,* $J = 2$), la méthode de prétraitement Conditional Info transforme chaque instance X_i ($i \in \{1, \ldots, N\}$) en un vecteur $\hat{X}_i$ de $2 \times d$ composantes de la manière suivante :

$$\hat{X}_i = (log(P(X_{i1} \in I_k|C_1), log(P(X_{i1} \in I_k|C_2), \ldots, log(P(X_{id} \in I_k|C_1), log(P(X_{id} \in I_k|C_2))$$

avec I_k ($k \in \{1, \ldots, t\}$) représente l'intervalle de discrétisation obtenu dans la première étape de Conditional Info.

Soit $D = \{(X_i, Y_i)\}_1^N$ un ensemble d'apprentissage de taille N, avec $X_i = \{X_{i1}, \ldots, X_{id}\}$ est un vecteur de d variables et $Y_{i \in \{1,\ldots,N\}} \in \{C_1, \ldots, C_J\}$ est une variable cible composée de J classes. Le prétraitement Conditional Info permet d'écrire une distance bayésienne dépendante de la classe, notée $dist_B^p$, pour la norme ℓ_p. La formule mathématique de cette distance entre deux instances $\hat{X}_1$ et $\hat{X}_2$ est définie de la manière suivante :

$$dist_B^p(\hat{X}_1, \hat{X}_2) = \sum_{j=1}^{J} \left\| log(P(\hat{X}_1|C_j)) - log(P(\hat{X}_2|C_j)) \right\|_p \tag{3.2}$$

avec $\|.\|_p$ est la distance de Minkowski :

$$\left\| log(P(\hat{X}_1|C_j)) - log(P(\hat{X}_2|C_j)) \right\|_p = \sqrt[p]{\sum_{m=1}^{d} \left| log(P(X_{1m}|C_j)) - log(P(X_{2m}|C_j)) \right|^p} \tag{3.3}$$

Il est facile de montrer que $dist_B^p$ est bien une distance. Elle vérifie les trois propriétés, à savoir, la séparation, la symétrie et l'inégalité triangulaire :

1. La séparation : $\forall(\hat{X}_1, \hat{X}_2) \in \mathcal{D}\ dist_B^p(\hat{X}_1, \hat{X}_2) = 0 \Leftrightarrow \hat{X}_1 = \hat{X}_2$
2. La symétrie : $\forall(\hat{X}_1, \hat{X}_2) \in \mathcal{D}\ dist_B^p(\hat{X}_1, \hat{X}_2) = dist_B^p(\hat{X}_2, \hat{X}_1)$
3. L'inégalité triangulaire : $\forall(\hat{X}_1, \hat{X}_2, \hat{X}_3) \in \mathcal{D}\ dist_B^p(\hat{X}_1, \hat{X}_3) \leq dist_B^p(\hat{X}_1, \hat{X}_2) + dist_B^p(\hat{X}_2, \hat{X}_3)$

• Pour **la norme** ℓ_1, la distance $dist_B^1(\hat{X}_1, \hat{X}_2)$, s'écrit comme suit :

$$dist_B^1(\hat{X}_1, \hat{X}_2) = \sum_{j=1}^{J} \sum_{m=1}^{d} \left| log(P(X_{1m}|C_j)) - log(P(X_{1m}|C_j)) \right| \tag{3.4}$$

• Pour **la norme** ℓ_2, la distance $dist_B^2(\hat{X}_1, \hat{X}_2)$, s'écrit comme suit :

$$dist_B^2(\hat{X}_1, \hat{X}_2) = \sum_{j=1}^{J} \sqrt{\sum_{m=1}^{d} \left| log(P(X_{1m}|C_j)) - log(P(X_{1m}|C_j)) \right|^2} \tag{3.5}$$

Pour chaque variable, la distance $dist_B^p$, peut s'interpréter comme une distance entre les ratios des probabilités ($log(P) - log(P') = log(\frac{P}{P'})$) en donnant une plus grande importance aux faibles différences de ratio (en raison de l'utilisation du logarithme).

Après avoir défini la distance dépendante de classe, on cherche à ce stade, à savoir si la distance obtenue permet de vérifier que "deux instances proches au sens de leur distribution (similarité) sont également proches au sens de leur comportement vis-à-vis de la classe à prédire". Il s'agit ici de montrer que la distance entre les distributions des classes conditionnellement aux données est

inférieure ou égale à la distance $dist_B^p$ entre les instances. Par conséquent, plus deux instances sont proches en termes de la distance $dist_B^p$ plus la probabilité qu'elles appartiennent de la même classe sera grande.

Adoptons le principe du prédicteur Bayésien, la distance entre les distributions des classes prédites pour deux instances $\hat{X}_1$ et $\hat{X}_2$ peut s'écrire selon la formule suivante :

$$\Delta^p(\hat{X}_1, \hat{X}_2) = \sum_{j=1}^{J} \left\| log(P(C_j|\hat{X}_1)) - log(P(C_j|\hat{X}_2)) \right\|_p \tag{3.6}$$

avec, $\forall i \in \{1, \ldots, N\}$

$$P(C_j|\hat{X}_i) = \frac{P(C_j)P(\hat{X}_i|C_j)}{P(\hat{X}_i)} = \frac{P(C_j)P(\hat{X}_{i1}, \ldots, \hat{X}_{id}|C_j)}{P(\hat{X}_i)} \tag{3.7}$$

En tenant compte de l'hypothèse d'indépendance des variables explicatives conditionnellement à la variable cible, l'équation 3.7 peut s'écrire de la manière suivante :

$$P(C_j|\hat{X}_i) = \frac{P(C_j)P(\hat{X}_i|C_j)}{P(\hat{X}_i)} = \frac{P(C_j)\prod_{m=1}^{d} P(X_{im}|C_j)}{P(\hat{X}_i)} \tag{3.8}$$

Par conséquent, on a :

$$log(P(C_j|\hat{X}_i)) = \sum_{m=1}^{d} log(P(X_{im}|C_j)) + log(P(C_j)) - log(P(\hat{X}_i)) \tag{3.9}$$

Partant de la définition donnée par l'équation 3.6 de la distance entre les distributions de classes prédites, on trouve la majoration suivante :

$$\Delta^p(\hat{X}_1, \hat{X}_2) \leq \left[dist_B^p(\hat{X}_1, \hat{X}_2) + J \left\| log(P(\hat{X}_2)) - log(P(\hat{X}_1)) \right\|_p \right] \tag{3.10}$$

Démonstration

$\Delta^p(\hat{X}_1, \hat{X}_2) = \sum_{j=1}^{J} \left\| log(P(C_j|\hat{X}_1)) - log(P(C_j|\hat{X}_2)) \right\|_p$

D'après l'équation 3.7, on trouve que

$$= \sum_{j=1}^{J} \left\| log(\tfrac{P(\hat{X}_1|C_j)P(C_j)}{P(\hat{X}_1)}) - log(\tfrac{P(\hat{X}_2|C_j)P(C_j)}{P(\hat{X}_2)}) \right\|_p$$

$$= \sum_{j=1}^{J} \left\| log(P(\hat{X}_1|C_j)) + log(P(C_j)) - log(P(\hat{X}_1)) - log(P(\hat{X}_2|C_j)) - log(P(C_j)) + log(P(\hat{X}_2)) \right\|_p$$

D'où

$$\leq \sum_{j=1}^{J} \left[\left\| log(P(\hat{X}_1|C_j)) - log(P(\hat{X}_2|C_j)) \right\|_p + \left\| log(P(\hat{X}_2)) - log(P(\hat{X}_1)) \right\|_p \right]$$

$$= \sum_{j=1}^{J} \left\| log(P(\hat{X}_1|C_j)) - log(P(\hat{X}_2|C_j)) \right\|_p + \sum_{j=1}^{J} \left\| log(P(\hat{X}_2)) - log(P(\hat{X}_1)) \right\|_p$$

$$= \sum_{j=1}^{J} \left\| log(P(\hat{X}_1|C_j)) - log(P(\hat{X}_2|C_j)) \right\|_p + J \left\| log(P(\hat{X}_2)) - log(P(\hat{X}_1)) \right\|_p$$
$$= dist_B^p(\hat{X}_1, \hat{X}_2) + J \left\| log(P(\hat{X}_2)) - log(P(\hat{X}_1)) \right\|_p$$

D'où

$$\Delta^p(\hat{X}_1, \hat{X}_2) \leq dist_B^p(\hat{X}_1, \hat{X}_2) + J \left\| log(P(\hat{X}_2)) - log(P(\hat{X}_1)) \right\|_p$$

Cette majoration signifie que deux instances de même probabilité globale proches au sens de $dist_B^1$ seront proches au sens de la prédiction des probabilités par classe cible. Elles seront également proches attribut par attribut, ce qui accroît la validité sémantique de la notion de proximité sous-jacente. Cette majoration est vraie aussi dans le cadre de la régression linéaire.

Note : Au cours de cette thèse, l'algorithme des K-moyennes est exécuté en utilisant la distance $dist_B^2$ en norme ℓ_2 et en utilisant la moyenne pour la mise à jour des centres.

Exemple illustratif

La figure 3.8 (partie gauche) présente le jeu de données Mouse qui est caractérisé par la présence de trois classes à prédire (noire, rouge et verte), deux variables descriptives continues (Variable 1 et Variable 2) et 490 instances. La première étape du prétraitement Conditional Info consiste à découper le domaine des deux variables descriptives en un nombre fini d'intervalles (5 intervalles pour la variable 1 et 3 intervalles pour la variable 2 comme le montre le tableau 3.2). Pour l'étape de recodage, chaque variable de X_i est transformée en 3 nouvelles variables synthétiques (puisque $|J| = 3$). Par exemple, la variable 1 d'une instance X_i ayant sa valeur dans l'intervalle $]0.27; 0.32]$ est transformée ainsi : $log(P(X_{i1} \in]0.27; 0.32]|noire), log(P(X_{i1} \in]0.27; 0.32]|rouge), log(P(X_{i1} \in]0.27; 0.32]|verte)$.

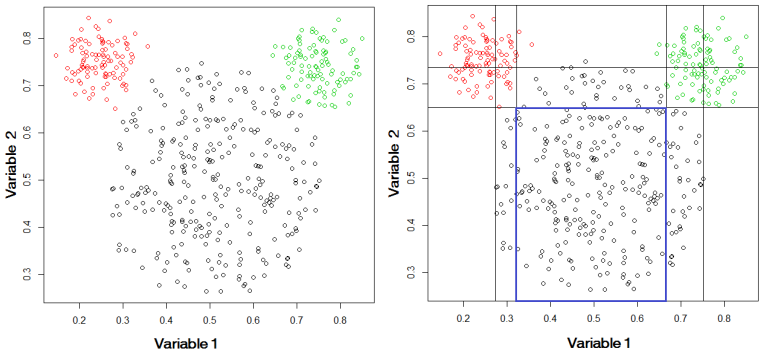


FIGURE 3.8 – Le jeu de données Mouse

A partir de ce type de recodage, on constate que toutes les instances ayant des valeurs dans les mêmes intervalles de discrétisation vont recevoir la même quantité d'information. A titre d'exemple, les instances entourées par le carré bleu dans la figure 3.8 (partie droite) ont bien des valeurs appartenant à l'intervalle de discrétisation $]0.32; 0.67]$ pour la variable 1 et des valeurs appartenant à l'intervalle de discrétisation $]-\inf; 0.65]$ pour la variable 2. Ces instances appartenant à la classe noire vont donc recevoir la même quantité d'information variable par variable. La figure 3.9 présente une illustration de la répartition des valeurs de la variable 1

Variable 1	Classe 'noire'	Classe 'rouge'	Classe 'verte'
$]-\inf;0.27]$	0	73	0
$]0.27;0.32]$	16	25	0
$]0.32;0.67]$	238	2	2
$]0.67;0.75]$	36	0	49
$]0.75;+\inf]$	0	0	49

Variable 2	Classe 'noire'	Classe 'rouge'	Classe 'verte'
$]-\inf;0.65]$	257	0	0
$]0.65;0.74]$	32	34	45
$]0.74;+\inf]$	1	66	55

TABLE 3.2 – La discrétisation des deux variables descriptives en utilisant l'approche MODL

avant et après le prétraitement. La partie gauche de cette figure présente la répartition des valeurs de départ de la variable 1. Tandis que les trois graphiques restant présentent la répartition des valeurs des trois variables synthétique obtenues après le prétraitement. Par exemple, le graphique présenté dans la partie droite de la figure 3.9 présente la répartition des valeurs de la variable $log(P(X_{i1} \in I_k|la\ classe\ verte))\ \forall i \in N$ avec I_k présente le nombre d'intervalles de discrétisation obtenu. d'après le tableau de droite présenté dans Table 3.2, on constate que, pour la variable 1, les instances de la classes verte sont réparties seulement dans 3 intervalles comme le prouve le graphique présenté dans la partie droite de la figure 3.9. Ceci prouve que les instances proches en termes de la distance $dist^1_B$ proposée seront proches au sens de la prédiction des probabilités par classe cible.

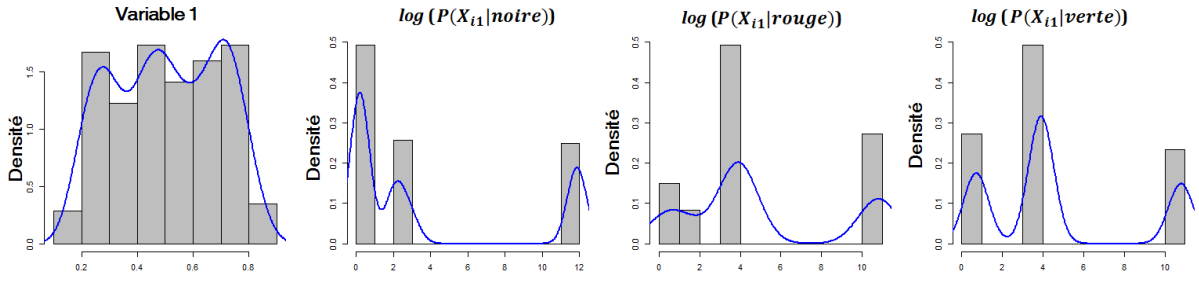


FIGURE 3.9 – La répartition de la variable 1 avant et après le prétraitement

Avantages

— Le premier point remarquable du prétraitement Conditional Info est sa capacité à construire une distance bayésienne qui vérifie que deux instances proches en termes de cette distance sont également proches en termes de leur probabilité d'appartenir à la même classe. Ce point pourrait être très utile pour l'algorithme des K-moyennes standard : l'incorporation de l'information cible dans les données sans la nécessité de modifier la fonction de coût de l'algorithme qui pourrait augmenter sa complexité algorithmique.

— Lors de la présence du bruit dans les données, après l'étape de discrétisation ou de groupage supervisé des variables, il existe deux possibilités : *i*) soit les exemples aberrants les plus proches reçoivent la même quantité d'information en formant un ou des groupes compacts (les points aberrants blues encadrés par le cadre magenta de la figure 3.10). Il est à signaler que ces points aberrants bleus sont bien de la classe rouge). *ii*) soit les exemples aberrants reçoivent la même quantité d'information que les instances qui forment le block de discrétisation ou de groupage (les points aberrants et les points rouges encadré par le cadre cyan de la figure 3.10). Par conséquent, l'effet de ces derniers sur ce groupe sera éliminé. Ceci montre que le prétraitement supervisé Conditional Info diminue en quelque sorte l'impact du bruit sur la qualité des résultats.

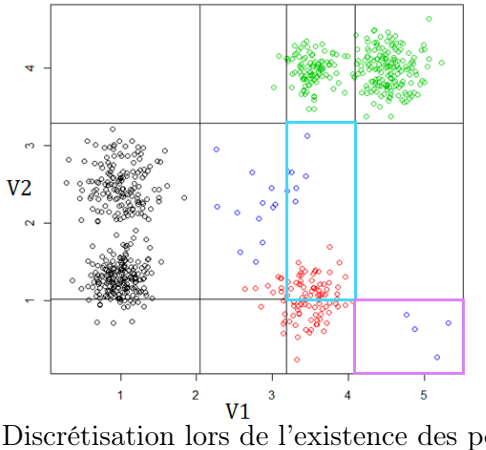


FIGURE 3.10 – Discrétisation lors de l'existence des points aberrants

Limites

Le prétraitement Conditional Info ne conserve pas la notion d'instance, c'est-à-dire que les instances peuvent recevoir la même quantité d'information (variable par variable) si toutes les variables de ces instances ont des valeurs qui appartiennent aux mêmes intervalles de discrétisation (par exemple, les instances entourées par le carré bleu dans la partie droite de la figure 3.8). Or ceci n'est pas un problème dans notre cas puisqu'on s'intéresse plutôt à faire des interprétations locales des données au lieu d'une interprétation individuelle (en se basant sur le prototype de chaque groupe). Les intervalles de discrétisation et les groupages en modalités ont dans notre cas une grande importance pour la réalisation de ces interprétations locales (voir l'exemple illustratif dans la section 3.5.3).

3.3 Protocole expérimental

3.3.1 Protocole

Comme il est connu, l'algorithme des K-moyennes est l'un des algorithmes de partitionnement qui converge rarement vers un optimum global. De ce fait, les résultats présentés dans cette section sont obtenus lorsque l'algorithme des K-moyennes est exécuté 100 fois avec différentes initialisations des centres en utilisant l'algorithme K-means++ [15]. La partition finale générée par l'algorithme des K-moyennes est choisie parmi les 100 partitions en utilisant un critère prédéterminé (voir le choix de la meilleure partition ci-dessous). Le reste des points à prendre en considération pour pouvoir aboutir aux résultats présentés dans la section 3.4 sont présentés ci-dessous.

- Les méthodes de prétraitement : Afin d'étudier l'impact de l'utilisation des méthodes de prétraitements supervisés (Conditional Info et Binarization) sur la qualité des résultats issus de l'algorithme des K-moyennes classique en termes de prédiction, nous allons les comparer aux méthodes de prétraitements usuels (non supervisés) pour les K-moyennes. Dans cette étude expérimentale, nous n'avons pas choisi de comparer les méthodes proposées avec les méthodes de prétraitement supervisées telles que l'analyse en composantes principales (ACP) en raison de la nécessité d'avoir des méthodes interprétables.

Pour les variables continues, selon notre connaissance, la normalisation des données est le pré-

traitement le plus communément utilisé dans la littérature pour l'algorithme des K-moyennes. Il permet d'ajuster une série de valeurs suivant une fonction de transformation pour les rendre comparables. La normalisation est nécessaire quand l'incompatibilité des unités de mesures entre les variables peut affecter les résultats sans apporter d'interprétations claires. Pour une comparaison équitable entre les variables ce prétraitement s'avère nécessaire. Les trois types de normalisation les plus répandus dans la littérature sont : *Min-Max Normalization (NORM)*, *Centrer et Réduire (CR)* et *Rank Normalization (RN)*.

1. **Min-Max Normalisation (NORM)** effectue une transformation linéaire sur les valeurs originelles des données. Si le minimum et le maximum de la variable u sont connus, alors cette dernière peut être transformée en une nouvelle variable qui prend ses valeurs dans $[0, 1]$. Cette transformation est effectuée en utilisant la formule suivante : $X'_{iu} = \frac{X_{iu} - min_{i=1,\dots,N} X_{iu}}{max_{i=1,\dots,N} X_{iu} - min_{i=1,\dots,N} X_{iu}}$ avec X_{iu} est la valeur d'origine de la variable u pour l'instance i.
2. **Centrer et réduire (CR)** fait référence à la transformation de données en soustrayant à chaque valeur la moyenne et en la divisant par l'écart-type. Cette transformation rendra toutes les valeurs en unités compatibles avec une distribution de moyenne 0 et d'écart-type 1. La formule qui permet cette transformation est : $X'_{iu} = \frac{X_{iu} - \mu}{\sigma}$ avec μ et σ sont respectivement la moyenne et l'écart type de la variable u.
3. **Rank Normalization (RN)** fait référence à la transformation de données en des groupes équitables de valeurs en respectant la répartition des valeurs de la variable u dans l'espace. Cette transformation commence par trier les valeurs de la variable u en ordre croissant. Ensuite, le vecteur résultant est divisé en H intervalles, où H représente le nombre d'intervalles. Suivant cet ordre, l'approche assigne à chaque intervalle un label $r \in \{1, \dots, H\}$. Finalement, pour la valeur X_{iu} appartient à l'intervalle r, alors celle-ci est recodée de la manière suivante $X'_{iu} = \frac{r}{H}$.

Pour les variables catégorielles, l'approche **Basic-Grouping-Binarization (BGB)** est la méthode de prétraitement la plus répandue dans la littérature. Cette approche vise à transformer les modalités des variables catégorielles en des valeurs booléennes. Les différentes étapes de **BGB** sont : *i*) grouper les modalités de chaque variable en g groupes de même fréquence où g est un paramètre utilisateur. *ii*) assigner à chaque groupe un label $v \in \{1, \dots, g\}$. *iii*) utiliser le codage disjonctif complet.

Le tableau 3.3 présente l'ensemble des prétraitements supervisés et non supervisés utilisé dans la partie expérimentale (Section 3.4).

Les prétraitements non supervisés			**Les prétraitements supervisés**		
Nom	variables numériques	variables catégorielles	Nom	variables numériques	variables catégorielles
RN-BGB	RN	BGB	BIN-BIN	BIN	BIN
CR-BGB	CR	BGB	CI-CI	CI	CI
NORM-BGB	NORM	BGB			

TABLE 3.3 – Liste des prétraitements utilisés

- Les jeux de données : Pour évaluer et comparer les différentes méthodes de prétraitements en fonction de leur capacité à aider l'algorithme des K-moyennes standard à atteindre l'objectif du clustering prédictif (première type), nous allons effectuer des tests sur différents jeux de données de l'UCI [1]. Ces jeux de données ont été choisis afin d'avoir des bases de données diverses en termes de nombre de classes J, de variables (continues M_n et/ou catégorielles M_c) et

d'instances N. Le tableau 3.5 présente l'ensemble des jeux de données utilisé dans la première partie d'expérimentation, tandis que, le tableau 3.4 présente l'ensemble des jeux de données utilisé dans la deuxième partie d'expérimentation. Ces derniers sont les jeux de données utilisés par Eick et al. dans [46] et par Al-Harbi et al. dans [6]. Pour une comparaison équitable entre les performances des algorithmes, ces jeux de données sont modifiés de la même façon que dans [46] et [6].

ID	**Nom**	M_n	M_c	N	J	J_{maj}
18	Iris	4	0	150	3	33
19	Pima	8	0	768	2	65
20	Auto-Import	15	11	205	2	60
21	Contraceptive	2	7	1473	2	61
22	Heart	10	3	270	2	56

TABLE 3.4 – Liste des jeux de données utilisés dans la deuxième partie d'expérimentation - (J_{maj} représente ≈ pourcentage classe majoritaire)

ID	**Nom**	M_n	M_c	N	J	J_{maj}
1	Wine	13	0	178	3	40
2	Glass	10	0	214	6	36
3	Horsecolic	7	20	368	2	63
4	Soybean	0	35	376	19	14
5	Breast	9	0	683	2	65
6	Australian	14	0	690	2	56
7	Vehicle	18	0	846	4	26
8	Tictactoe	0	9	958	2	65
9	LED	7	0	1000	10	11
10	German	24	0	1000	2	70
11	Segmentation	19	0	2310	7	14
12	Abalone	7	1	4177	28	16
13	Waveform	21	0	5000	3	34
14	Adult	7	8	48842	2	76
15	Mushroom	0	22	8416	2	53
16	PenDigits	16	0	110992	10	10
17	Phoneme	256	0	2254	5	26

TABLE 3.5 – Liste des jeux de données utilisés dans la première partie d'expérimentation - (J_{maj} représente ≈ pourcentage classe majoritaire)

- Le choix de la meilleure partition : Lorsque l'algorithme des K-moyennes est exécuté plusieurs fois (100 fois dans cette étude expérimentale), le choix de la meilleure partition est alors une étape cruciale. Dans cette étude expérimentale, la meilleure partition est définie comme étant la partition qui minimise l'erreur quadratique moyenne (MSE). La formule mathématique utilisée pour la MSE est donnée comme suit :

$$MSE = \frac{1}{N}\frac{1}{Z}\frac{1}{K}\sum_{i=1}^{N}\sum_{z=1}^{Z}\sum_{t=1}^{K}(XR_i^z - k_t^z)^2 \tag{3.11}$$

— N est le nombre d'instances dans l'ensemble de données.

— Z est le nombre de variable après le processus de prétraitement. Par exemple, pour conditional Info, $Z = (M_n + M_c) \times J$.
— K est le nombre de clusters.
— XR est le nouveau vecteur d'instance après le processus de prétraitement utilisé. Par exemple, pour conditional Info, ce nouveau vecteur XR est de dimension $(M_n + M_c) \times J$.
— k_t^z est le centre de gravité du cluster t, représenté sous forme d'un un vecteur de dimension Z.

- Le nombre de clusters (K) : Étant donné un prétraitement i, le nombre de clusters varie de J (nombre de classes) jusqu'à K_i. Pour chaque jeu de données, K_i a été déterminé au préalable de manière à ce que la partition obtenue, avec $K{=}K_i$ permette d'obtenir un ratio (inertie inter / inertie totale) de 80%. La valeur de K_i ($i \in \{CI-CI, BIN-BIN, RN-BGB, CR-BGB, NORM-BGB\}$) pour chaque jeu de données et pour chaque prétraitement i est indiquée dans le tableau 3.6. Il est à noter que dans cette étude, le nombre de clusters K ne doit pas être inférieur à C puisqu'on suppose que la variable cible a une structure interne à découvrir. Pour plus de détails voir Annexe A de ce mémoire.

Données	**CI-CI**	**BIN-BIN**	**RN-BGB**	**CR-BGB**	**NORM-BGB**
Wine	12	47	38	35	33
Glass	15	21	25	17	12
Horsecolic	6	7	200	11	14
Soybean	20	20	49	49	49
Breast	4	56	12	15	12
Australian	22	74	210	58	126
Vehicle	11	126	24	17	16
Tictactoe	12	13	496	499	500
LED	17	19	19	17	19
German	7	10	363	280	217
Segmentation	23	64	21	15	8
Abalone	29	29	29	29	29
Waveform	86	64	64	64	64
Adult	12	64	64	64	64
Mushroom	8	78	64	250	64
PenDigits	73	64	33	28	22
Phoneme	64	64	64	64	64
Iris	3	6	4	4	4
Pima	10	22	74	69	61
Auto-imports	5	9	33	19	35
Contraceptive	6	18	92	56	73
Heart	23	29	90	67	53

TABLE 3.6 – Détermination de K_i pour chaque prétraitement i

- L'attribution des classes aux groupes appris : A la fin du processus d'apprentissage, chaque groupe appris prend j comme étiquette si la majorité des exemples qui le forme sont de la classe j (*i.e.,* l'utilisation du vote majoritaire).

- La prédiction : A la présence d'une nouvelle instance, l'algorithme lui affecte l'étiquette du cluster qui lui est plus proche [3] (*i.e.,* l'utilisation du 1 plus proche voisin).

3. Une instance i est plus proche au cluster C_1 que au cluster C_2 si et seulement $dist(i, g_1) < dist(i, g_2)$ avec

- La cross validation : Pour la première partie d'expérimentation (Section 3.4.1), pour pouvoir comparer les résultats obtenus, un 2×5 folds cross validation a été effectué sur chaque jeu de données. Les résultats sont donc présentés comme une moyenne de 10 tests. Pour la deuxième partie expérimentale (Section 3.4.2), pour être en mesure de comparer nos résultats avec ceux de Eick [46] et de Al-Harbi [6], nous effectuons : *i*) un 20×5 folds cross validation (comme les expérimentations effectuées par Al-Harbi dans [6]) pour les jeux de données Auto-import, Breast, Contraceptive et Pima. Ces jeux de données sont également modifiés de la même façon que dans [6]. *ii*) un 10×10 folds cross validation (comme les expérimentations effectuées par Eick dans [46]) pour les jeux de données Glass, Heart, Vehicle et Iris.

3.3.2 Évaluation de la qualité du clustering prédictif

Les algorithmes du **clustering prédictif du premier type** privilégient principalement l'axe de description. Comme dans la classification supervisée, ces algorithmes cherchent à prédire correctement la classe des nouvelles instances. La seule différence ici est que les algorithmes du clustering prédictif du premier type génèrent des clusters souvent supérieur au nombre des classes. Pour l'évaluation des performances prédictives de ces algorithmes, on cherche souvent à comparer deux partitions ayant un nombre différent de groupes : la première partition est la partition contenant les varies classes des instances et la deuxième partition est celle qui contient les ID-clusters générés par les algorithmes du clustering prédictif.

Pour ce cadre d'étude, il est clair que l'utilisation d'un critère tel que l'accuracy s'avère insuffisant. Parmi les nombreux critères existant dans la littérature permettant de comparer deux partitions ayant un nombre de clusters différents, on avons choisi d'utiliser : l'indice de rand ajusté (ARI) [57] et la variation d'information [76]. Ce choix a été basé d'après l'étude réalisée dans [102].

Indice de Rand Ajusté (ARI)

Soit $D = \{(X_i, Y_i)\}_1^N$ un ensemble d'apprentissage de taille N et $\mathcal{C}_1 = \{s_1, \ldots, s_{K_1}\}$ et $\mathcal{C}_2 = \{u_1, \ldots, u_{K_2}\}$ deux partitions ayant respectivement K_1 et K_2 clusters tel que $D = \cup_{i=1}^{K_1} s_i = \cup_{j=1}^{K_2} u_j$. L'indice de rand ajusté (ARI) [57] permettant de comparer les deux partitions $\mathcal{C}_1$ et $\mathcal{C}_2$ est donné par la formule suivante :

$$ARI = \frac{\sum_{i,j}\binom{N_{ij}}{2} - \left[\sum_i \binom{N_{i.}}{2} \sum_j \binom{N_{.j}}{2}\right] / \binom{N}{2}}{\frac{1}{2}\left[\sum_i \binom{N_{i.}}{2} + \sum_j \binom{N_{.j}}{2}\right] - \left[\sum_i \binom{N_{i.}}{2} \sum_j \binom{N_{.j}}{2}\right] \binom{N}{2}} \tag{3.12}$$

avec

— N_{ij} représente le nombre d'instances appartenant à la fois au cluster s_i et au cluster u_j .

— $N_{i.}$ représente le nombre d'instances appartenant au cluster s_i $\forall i \in \{1, \ldots, K_1\}$.

— $N_{.j}$ représente le nombre d'instances appartenant au cluster u_j $\forall j \in \{1, \ldots, K_2\}$.

L'indice de rand ajusté (ARI) est compris entre 0 et 1. Il est égal à 1 lorsque deux partitions sont exactement identiques. c'est critère à maximiser

g_1 (respectivement g_2) est le centre de gravité du cluster C_1 (respectivement C_2).

Variation d'Information (VI)

Le critère de comparaison Variation d'Information (VI) issu de la théorie de l'information, quantifie l'information apportée par la connaissance d'une partition $\mathcal{C}$ sur une partition $\mathcal{C}'$. Soit N_j le cardinal de la classe c_j. Soit :

- $P(j) = \frac{N_j}{N}$ la probabilité d'une observation X_i choisie au hasard appartienne à la classe c_j.
- $P(j,l) = \frac{|c_j \cap c'_j|}{N}$ la probabilité que les observations appartiennent aux classes $c_j \in \mathcal{C}$ et $c'_j \in \mathcal{C}'$.

La Variation d'Information entre deux partitions $\mathcal{C}$ et $\mathcal{C}'$ est la somme de l'information sur $\mathcal{C}$ que l'on perd et de l'information sur $\mathcal{C}'$ que l'on gagne lorsqu'on passe de la partition $\mathcal{C}$ à la partition $\mathcal{C}'$. Formellement, on a :

$$VI(\mathcal{C},\mathcal{C}') = \mathcal{H}(\mathcal{C}) + \mathcal{H}(\mathcal{C}') - 2\mathcal{I}(\mathcal{C},\mathcal{C}') \tag{3.13}$$

$\mathcal{H}$ et $\mathcal{I}$ présentent respectivement l'entropie et l'information mutuelle. Pour plus de détails sur cette mesure, voir [76]. Une version normalisée de VI, notée VIn a été proposé dans [102]. Cette dernière est donnée par l'équation 3.14 :

$$VIn(\mathcal{C},\mathcal{C}') = 1 - \frac{2\ \mathcal{I}(\mathcal{C},\mathcal{C}')}{\mathcal{H}(\mathcal{C}) + \mathcal{H}(\mathcal{C}')} \tag{3.14}$$

VIn est compris entre 0 et 1. Elle est égale à zéro si et seulement si $\mathcal{C}$ et $\mathcal{C}'$ sont identiques. Au cours de cette thèse, la version normalisée VI est utilisée que l'en note également VI.

Pour évaluer la performance prédictive des modèles utilisés dans ce chapitre, nous allons utiliser l'indice de rand ajusté (ARI).

Dans le cadre du **clustering prédictif du deuxième type**, on cherche à réaliser un compromis entre la prédiction et la description. À notre connaissance, il n'existe pas dans la littérature de critère global qui permette de mesurer ce compromis. Une possibilité pour évaluer ce compromis est d'utiliser le Front de Pareto (pour plus de détails, voir [20]). Cependant, pour l'axe de description, les résultats issus de l'algorithme des K-moyennes en utilisant les différents prétraitements ne sont pas comparables. En effet, le nombre de variables ainsi que la plage de variation diffèrent d'un prétraitement à l'autre. Par conséquent, l'utilisation d'un critère d'évaluation interne tel que Davies-Bouldin [38] ou la MSE (équation 4.2) ne permet pas de réaliser une telle comparaison. De ce fait, on suppose dans cette étude expérimentale que la partie "description" est garantie par l'algorithme des K-moyennes [4] et on évalue seulement la partie "prédiction" en utilisant le critère d'évaluation communément utilisé dans la littérature à savoir, *indice de rand ajusté ARI* (ou Adjusted Rand Index).

En dehors de ces deux critères, d'autres critères ont également été utilisés dans cette thèse tels l'erreur quadratique moyenne "MSE" (voir page 60) , la précision "ACC" (voir page 69) et BACC (Balanced Accuracy) (voir page 67).

3.4 Résultats

Pour être en mesure de répondre à la question posée dans la section 4.1, à savoir : "*les méthodes de prétraitement supervisées pourrait-elles aider l'algorithme des K-moyennes standard*

4. Dans la phase d'apprentissage, la partition finale générée par l'algorithme des K-moyennes est définie comme étant la partition ayant la meilleure MSE (parmi les 100 partitions, voir le choix de la meilleure partition).

à fournir de bons résultats au sens du clustering prédictif ?", nous allons diviser notre étude expérimentale en deux grandes parties.

Dans la première partie (Section 3.4.1), nous allons comparer les méthodes de prétraitement usuelles (non supervisées) pour les K-moyennes avec les deux méthodes de prétraitements supervisées (Conditional Info et Binarization) proposées dans les deux sections 3.2.2 et 3.2.3. Cette partie a pour but d'étudier l'impact de l'utilisation des méthodes de prétraitements supervisées sur la qualité des résultats issus par l'algorithme classique des K-moyennes. Il est à rappeler que la qualité discutée dans ce chapitre est définie comme étant le pouvoir prédictif de l'algorithme à bien prédire la classe des nouvelles instances. Ce pouvoir est calculé à l'aide de l'indice de rand ajusté ARI (équation 3.12).

Dans la deuxième partie (Section 3.4.2), nous allons comparer les performances prédictives de l'algorithme des K-moyennes précédé à chaque fois par un prétraitement supervisé avec les deux algorithmes les plus répandus dans le cadre du clustering supervisé. Ces algorithmes sont notamment, l'algorithme de Eick et al. proposé dans [46] et l'algorithme de Al-Harbi et al. proposé dans [6]. Cette partie a pour but d'étudier le degré de compétitivité de l'algorithme classique des K-moyennes précédé par les deux prétraitements supervisés (Conditional Info, et Binarization) avec ces deux algorithmes de clustering supervisés.

3.4.1 Distances supervisées *Vs.* distances non supervisées

Le but de cette première étude expérimentale est de vérifier si l'incorporation de l'information cible dans les données via un prétraitement supervisé pourrait aider l'algorithme des K-moyennes standard à atteindre l'objectif du clustering prédictif. Comme défini dans la section 1.6 du Chapitre 1, le clustering prédictif traite principalement trois axes, à savoir : la description, la prédiction et l'interprétation. Pour les différents prétraitements utilisés dans cette partie (voir Section 3.3), il s'avère difficile de comparer leurs performances suivant l'axe de description. En effet, À notre connaissance, les critères d'évaluation internes proposés dans le cadre du clustering sont tous basés sur une mesure de similarité. Or, ces méthodes de prétraitement n'ont pas forcément la même plage de variation ni le même nombre de variables. Pour cette raison, on suppose dans cette étude expérimentale que l'axe de description est garanti par l'algorithme des K-moyennes [5] et on évalue uniquement la performance des méthodes suivant l'axe de prédiction. Concernant l'axe d'interprétation, il sera discuté dans la section 3.5.3. Cette première partie d'expérimentation cherche donc à savoir si l'algorithme classique des K-moyennes précédé par les prétraitements supervisés parvient à bien prédire la classe des nouvelles instances comparé aux prétraitements non supervisés.

Pour l'algorithme des K-moyennes, le choix du nombre de clusters (K) est un problème en soi : il n'est pas évident de connaître à l'avance pour chaque jeu de données le nombre de clusters convenable. Cette étude expérimentale est donc divisée en deux selon la façon de choisir le nombre de clusters K. Dans la première partie, ce dernier est considéré comme un paramètre utilisateur : K est égal, pour chaque jeu de données, au nombre de classes (J) à prédire. Dans la deuxième partie le nombre de clusters est considéré comme une sortie de l'algorithme : pour chaque prétraitement i, l'algorithme des K-moyennes est exécuté avec différent nombre de clusters (de J jusqu'à K_i, pour plus de détails, voir le choix du nombre de clusters dans la section 3.3), ensuite, le nombre de clusters optimal $K_{opti} \in \{J, \ldots, K_i\}$ est considéré comme étant la

5. La partition finale générée par l'algorithme des K-moyennes est définie comme étant la partition qui optimise l'erreur quadratique moyenne (MSE) parmi les 100 partitions (voir le choix de la meilleure partition dans Section 3.3).

partition qui optimise l'indice de rand ajusté.

A. Le nombre de clusters est une entrée

Dans cette partie, on se limite au cas où le nombre de clusters K est égal au nombre de classes à prédire J. Dans ce cas, le problème du clustering prédictif devient un problème de classification supervisée. Le but ici est de connaître la capacité de l'algorithme classique des K-moyennes précédé par les méthodes de prétraitements supervisés à prédire correctement la classe des nouvelles instances.

La figure 3.11 présente les performances prédictives moyennes (en termes d'ARI) de l'algorithme des K-moyennes précédé par les différentes méthodes de prétraitement (supervisées et non supervisées) pour 17 jeux de données de l'UCI (voir tableau 3.5 de la section 3.3).

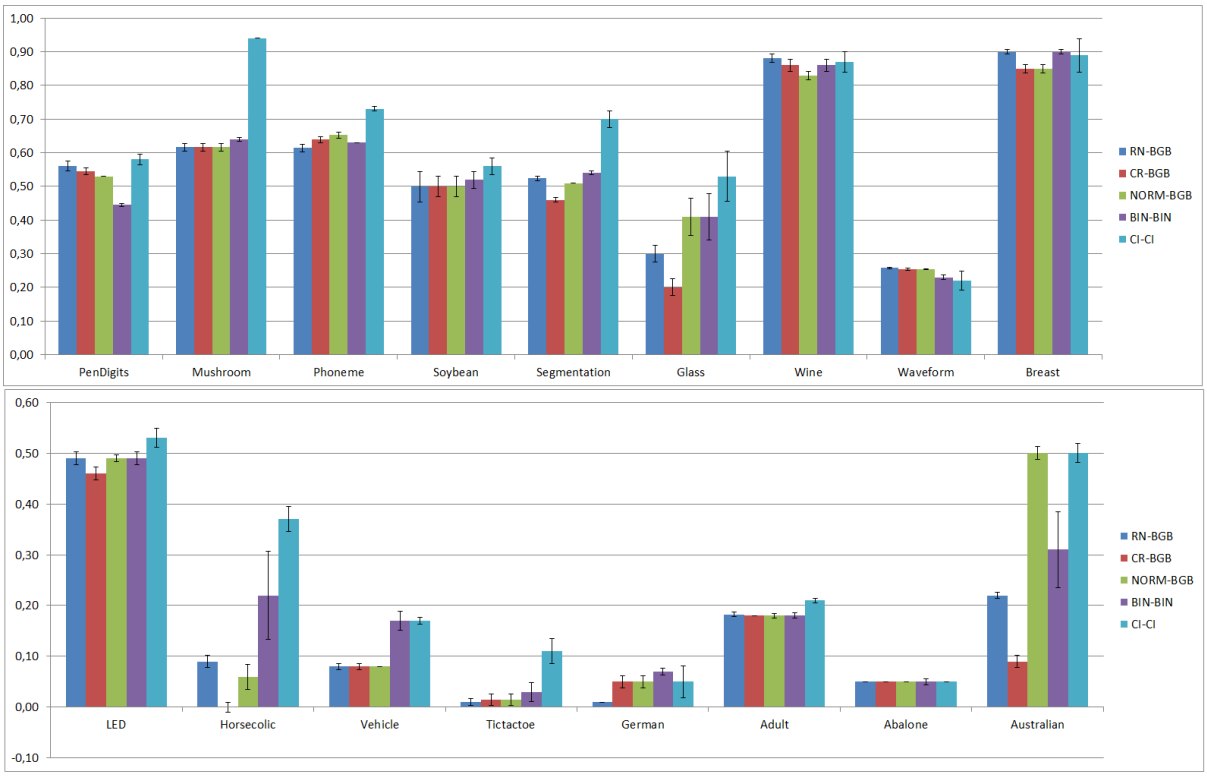


FIGURE 3.11 – Les performances prédictives moyennes des K-moyennes précédé par les différentes méthodes de prétraitement en utilisant l'ARI

Dans cette figure, on observe que la méthode de prétraitement supervisée "Conditional Info" a une performance prédictive soit : *i*) meilleure de celles de prétraitement non supervisés (12 jeux de données sur 17), *ii*) compétitive avec les performances de ces derniers (5 jeux de données sur 17). L'ensemble des tableaux contenant les résultats détaillés (en apprentissage et en test) qui servent à obtenir ces résultats synthétiques présentés dans cette partie sont situés dans la section B.2.1 l'annexe B.

À ce stade, pour être en mesure de classer les différentes méthodes de prétraitement selon leur pouvoir prédictif sur les 17 jeux de données, nous allons utiliser le test de Friedman couplé au test post-hoc de Nemenyi [41] (voir Section B.1 de l'annexe B). La figure 3.12 présente les résultats des comparaisons des performances prédictives en termes d'ARI en apprentissage (partie gauche de la figure) et en test (partie droite de la figure) de l'algorithme des K-moyennes en utilisant à chaque fois une méthode de prétraitement. Les méthodes sont classées par ordre décroissant selon leurs performances prédictives en se basant sur la moyenne des rangs : plus le rang de la méthode est proche de 1 meilleure est la prédiction.

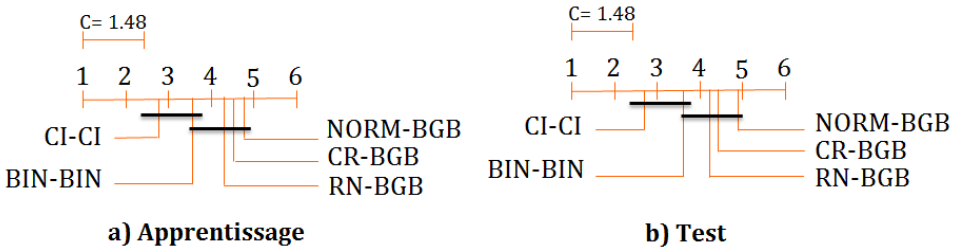


FIGURE 3.12 – Le test de Friedman couplé au test post-hoc de Nemenyi pour les 21 jeux de données en utilisant l'ARI en apprentissage a) et en test b)

D'après les résultats de test de Friedman, il existe une différence significative entre les 4 méthodes de prétraitement ($p_{value} < 10^{-4} \ll 0.05$). Que ce soit en apprentissage ou en test, d'après les résultats du test de Nemenyi, on constate que les deux méthodes supervisées sont celles qui ont une bonne performance en termes de prédiction tandis que la méthode Normalization est celle qui fournit des résultats moins bons en termes de prédiction.

B. Le nombre de clusters est une sortie

Dans le cadre du clustering prédictif, on s'attend à ce que le nombre de clusters soit supérieur au nombre de classes du fait qu'on souhaite découvrir à ce stade la structure interne de la variable cible (on suppose qu'au moins une des classes contient une structure sous-jacente à découvrir). Dans cette partie, on considère que le nombre de clusters K comme une sortie de l'algorithme des K-moyennes : pour chaque jeu de données et pour chaque prétraitement i, l'algorithme des K-moyennes est exécuté avec différentes valeurs de K (de J jusqu'à K_i) tout en effectuant une validation croisée en 10 folds. Ensuite, à la fin de la phase d'apprentissage, le nombre de clusters considéré est celui qui correspond à la partition ayant une bonne performance en termes de l'indice de rand ajusté (*i.e.,* celle qui optimise l'ARI). Puisque le critère d'ARI est utilisé pour sélectionner le nombre optimal de clusters, la qualité prédictive de l'algorithme en question précédé par les différentes méthodes de prétraitements est mesurée dans cette partie en utilisant "*Balanced Accuracy*"(BACC).

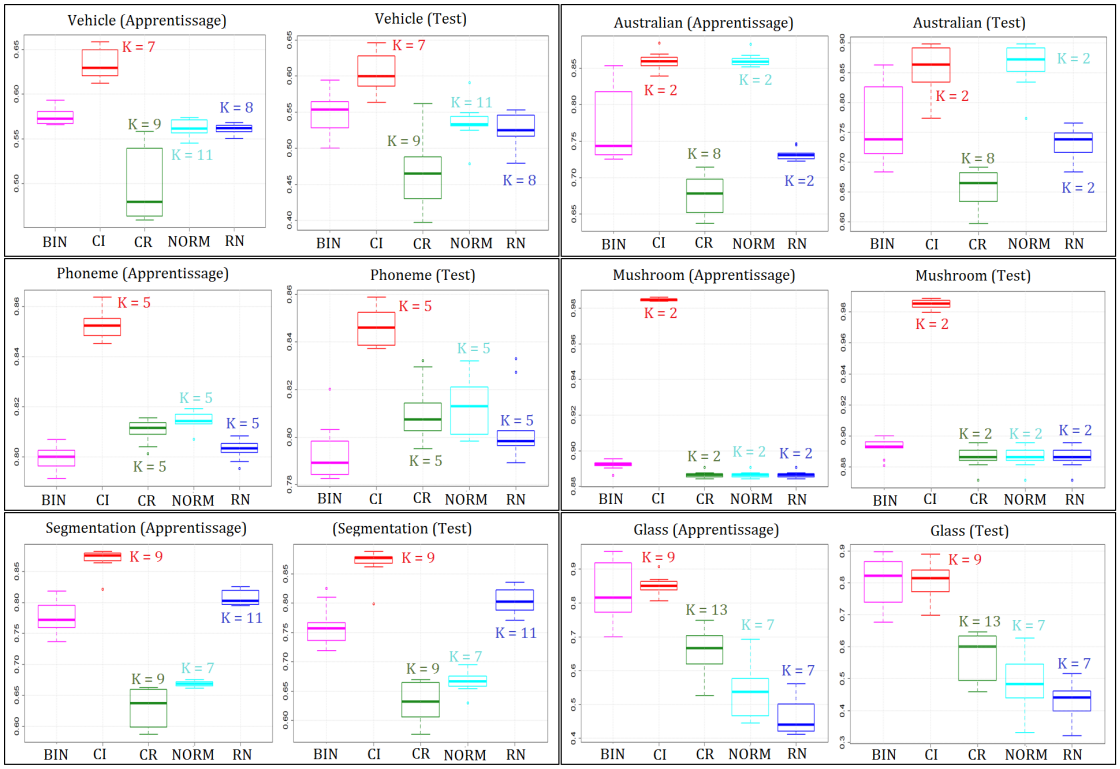


FIGURE 3.13 – La performances moyenne (en termes de BACC en test) de l'algorithme des K-moyennes standard précédé par les différentes méthodes de prétraitement dans le cas où K est une sortie.

La figure 3.13 et le tableau 3.7 présentent les performances prédictives moyennes (en termes de BACC) de l'algorithme classique des K-moyennes précédé par les différentes méthodes de prétraitement lorsque le nombre de clusters est considéré comme une sortie. Ces résultats montrent clairement que Conditional Info est la méthode qui fournit de bons résultats en termes de prédiction tout en gardant un nombre minimal de clusters.

Données	Méthodes	K	BACC (A)	BACC (T)	Données	Méthodes	K	BACC (A)	BACC (T)
German	RN-BGB	6	0.5 ± 0	0.5 ± 0	Horsecolic	RN-BGB	3	0.45 ± 0.01	0.58 ± 0.08
	NORM-BGB	2	0.5 ± 0	0.5 ± 0		NORM-BGB	6	0.77 ± 0.08	0.59 ± 0.19
	CR-BGB	2	0.5 ± 0	0.5 ± 0		CR-BGB	11	$\mathbf{0.94 \pm 0.04}$	0.67 ± 0.19
	BIN-BIN	2	0.5 ± 0	0.5 ± 0		BIN-BIN	2	0.50 ± 0.02	0.62 ± 0.07
	CI-CI	5	$\mathbf{0.56 \pm 0.02}$	$\mathbf{0.54 \pm 0.03}$		CI-CI	2	0.53 ± 0.01	$\mathbf{0.70 \pm 0.09}$
LED	RN-BGB	11	$\mathbf{0.71 \pm 0.02}$	0.7 ± 0.02	Soyeban	RN-BGB	22	0.74 ± 0.03	0.71 ± 0.04
	NORM-BGB	11	0.7 ± 0.02	0.69 ± 0.03		NORM-BGB	22	0.74 ± 0.03	0.71 ± 0.04
	CR-BGB	10	0.66 ± 0.02	0.66 ± 0.02		CR-BGB	22	0.74 ± 0.03	0.71 ± 0.04
	BIN-BIN	11	0.7 ± 0.02	0.69 ± 0.03		BIN-BIN	22	0.77 ± 0.02	0.75 ± 0.03
	CI-CI	10	$\mathbf{0.71 \pm 0.02}$	$\mathbf{0.71 \pm 0.02}$		CI-CI	20	$\mathbf{0.79 \pm 0.01}$	$\mathbf{0.79 \pm 0.02}$
Tictactoe	RN-BGB	17	$\mathbf{0.99 \pm 0.01}$	$\mathbf{0.99 \pm 0.01}$	Wine	RN-BGB	3	0.98 ± 0.01	0.97 ± 0.02
	NORM-BGB	17	$\mathbf{0.99 \pm 0.01}$	$\mathbf{0.99 \pm 0.01}$		NORM-BGB	3	0.96 ± 0.00	0.95 ± 0.02
	CR-BGB	19	$\mathbf{0.99 \pm 0.01}$	$\mathbf{0.99 \pm 0.01}$		CR-BGB	3	0.97 ± 0.01	0.96 ± 0.02
	BIN-BIN	8	0.66 ± 0.02	0.62 ± 0.06		BIN-BIN	3	0.97 ± 0.01	0.96 ± 0.02
	CI-CI	2	0.62 ± 0.08	0.62 ± 0.08		CI-CI	3	$\mathbf{0.98 \pm 0.01}$	$\mathbf{0.97 \pm 0.01}$
Adult	RN-BGB	2	0.5 ± 0.00	0.5 ± 0.00	Waveform	RN-BGB	5	0.74 ± 0.00	0.74 ± 0.01
	NORM-BGB	2	0.5 ± 0.00	0.5 ± 0.00		NORM-BGB	5	0.74 ± 0.00	0.74 ± 0.01
	CR-BGB	5	0.5 ± 0.00	0.5 ± 0.00		CR-BGB	5	0.74 ± 0.00	0.74 ± 0.01
	BIN-BIN	2	0.5 ± 0.00	0.5 ± 0.00		BIN-BIN	5	$\mathbf{0.75 \pm 0.01}$	$\mathbf{0.75 \pm 0.01}$
	CI-CI	4	$\mathbf{0.54 \pm 0.01}$	$\mathbf{0.54 \pm 0.01}$		CI-CI	4	0.59 ± 0.02	0.58 ± 0.02
PenDigits	RN-BGB	12	0.82 ± 0.01	0.82 ± 0.01	Breast	RN-BGB	2	0.98 ± 0.00	0.98 ± 0.01
	NORM-BGB	12	0.82 ± 0.00	0.81 ± 0.01		NORM-BGB	2	0.95 ± 0.01	0.95 ± 0.01
	CR-BGB	13	0.83 ± 0.00	0.83 ± 0.00		CR-BGB	2	0.95 ± 0.01	0.95 ± 0.01
	BIN-BIN	11	$\mathbf{0.97 \pm 0.00}$	$\mathbf{0.97 \pm 0.00}$		BIN-BIN	2	$\mathbf{0.98 \pm 0.01}$	$\mathbf{0.98 \pm 0.00}$
	CI-CI	12	0.76 ± 0.03	0.75 ± 0.03		CI-CI	2	$\mathbf{0.98 \pm 0.02}$	$\mathbf{0.98 \pm 0.00}$
Abalone	RN-BGB	28	0.08 ± 0.01	0.09 ± 0.00					
	NORM-BGB	28	0.11 ± 0.01	0.12 ± 0.01					
	CR-BGB	28	$\mathbf{0.12 \pm 0.01}$	$\mathbf{0.13 \pm 0.01}$					
	BIN-BIN	29	$\mathbf{0.12 \pm 0.01}$	$\mathbf{0.13 \pm 0.01}$					
	CI-CI	28	$\mathbf{0.12 \pm 0.01}$	$\mathbf{0.13 \pm 0.01}$					

TABLE 3.7 – Les performances moyennes (en termes de BACC) de l'algorithme des K-moyennes standard précédé par les différents méthodes de prétraitement dans le cas où K est une sortie (A : Apprentissage, T : Test)

3.4.2 Distances supervisées *Vs.* Clustering supervisé

Dans la littérature, plusieurs algorithmes de clustering ont été modifiés (comme l'algorithme des K-moyennes) dans le but de les adapter au problème de la classification supervisée. Ces algorithmes sont connus sous le nom de "clustering supervisé". Dans cette deuxième partie d'expérimentation, nous allons comparer les performances prédictives moyennes de l'algorithme classique des K-moyennes précédé à chaque fois par une méthode de prétraitement supervisée (Conditional Info et Binarization) avec les performance moyennes des deux algorithmes de clustering supervisé les plus répandus dans la littérature (algorithme de Eick *et al.* et algorithme de AL-Harbi *et al.*). Le but de cette partie est de savoir si la modification d'une seule étape de l'algorithme classique des K-moyennes (*i.e.,* l'étape de prétraitement des données, voir la section 1.6.3 du Chapitre 1) aboutit à être compétitif avec les algorithmes de clustering supervisé communément utilisés.

L'algorithme de AL-Harbi prend le nombre de clusters en entrée de l'algorithme. Pour chaque jeu de données, ce nombre est défini comme étant le nombre de classes à prédire. Les jeux de données utilisés sont modifiés de la même façon que AL-Harbi dans [6] pour une comparaison équitable des résultats. Le tableau 3.8 (partie en bas) présente les performances prédictives moyennes en termes d'accuracy (ACC) de l'algorithme de AL-Harbi et de l'algorithme des K-

Comparaison avec l'algorithme de Eick : (K en une sortie)						
	Glass		Heart		Iris	
	K	ACC en test	K	ACC en test	K	ACC en test
Eick algorithm	34	0.636	2	0.745	3	**0.973**
K-means with BIN	7	$\mathbf{0.677 \pm 0.091}$	2	$\mathbf{0.813 \pm 0.076}$	4	0.933 ± 0.064
K-means with C.I	6	0.620 ± 0.093	2	0.808 ± 0.079	3	0.902 ± 0.083
Comparaison avec l'algorithme de Al-Harbi : (K est une entrée)						
	Auto-import		Breast		Pima	
	K	ACC en test	K	ACC en test	K	ACC en test
Algorithme de Al-Harbi	2	**0.925**	2	**0.976**	2	**0.746**
K-moyennes avec BIN	2	0.831 ± 0.054	2	0.974 ± 0.012	2	0.699 ± 0.043
K-moyennes avec C.I	2	0.814 ± 0.102	2	0.969 ± 0.020	2	0.740 ± 0.033

TABLE 3.8 – Comparaison des prétraitements supervisés avec les deux algorithmes de Eick et de Al-Harbi

moyennes standard précédé à chaque fois par une méthode de prétraitement supervisé (CI et BIN). Ces résultats montrent que les performances de l'algorithme de K-moyennes précédé par les méthodes de prétraitement sont compétitives à celle obtenues par l'algorithme de Al-Harbi. Cependant, il est important de rappeler que ce dernier intègre un algorithme génétique dans le fonctionnement de l'algorithme des K-moyennes afin d'optimiser une fonction objectif prédéfinie l'auteur. Ceci augmente la complexité algorithmique de l'algorithme. De ce fait, l'utilisation de l'algorithme de K-moyennes standard précédé par une étape de prétraitement reste préférable.

Pour l'algorithme de Eick, le nombre de clusters est considéré comme une sortie de l'algorithme. Le tableau 3.8 (partie en haut) présente les performances prédictives moyennes en termes d'accuracy (ACC) de l'algorithme de Eick et de l'algorithme des K-moyennes standard précédé à chaque fois par une méthode de prétraitement supervisé (CI et BIN). Ces résultats montrent que les performances de l'algorithme de K-moyennes précédé par les méthodes de prétraitement supervisées sont compétitives à celle obtenues par l'algorithme de Eick. L'algorithme des K-moyennes avec les prétraitements supervisés conserve un nombre faible de clusters comparé à l'algorithme de Eick. De plus, l'algorithme de Eick est un algorithme qui nécessite beaucoup d'efforts de calcul (voir section 1.5.2 du Chapitre 1). De ce fait, l'utilisation de l'algorithme des K-moyennes standard précédé par une étape de prétraitement reste préférable.

3.4.3 Conclusion

Dans cette première partie d'expérimentation nous avons pu montrer que le prétraitement supervisé Conditional Info parvient à aider l'algorithme classique des K-moyennes dans le contexte supervisé comparé aux méthodes de prétraitement non supervisées usuelles pour les K-moyennes. De plus, avec seulement la modification d'une seule étape de l'algorithme classique des K-moyennes, ce dernier arrive à avoir des résultats très compétitifs en termes de prédiction avec les deux algorithmes les plus répandus dans le cadre du clustering supervisé (algorithme de Eick et algorithme de Al-Harbi).

3.5 Discussion

Comme nous l'avons évoqué précédemment, il est difficile de comparer la qualité des résultats (en termes de description) générés par les différentes méthodes de prétraitement : les données

résultants des différentes méthodes de prétraitement n'ont pas la même plage de variation ni le même nombre de variables. Or, la majorité des critères existant dans la littérature permettant d'évaluer l'axe de description se basent principalement sur une mesure de similarité.

Cette section a pour but d'évaluer principalement les deux axes non traités dans la section précédente, à savoir, l'axe de description et l'axe d'interprétation. Dans un premier temps, nous allons étudier dans la section 3.5.1 la capacité des différentes méthodes de prétraitement à avoir des présentations des données pertinentes (évaluée par le degré de chevauchement dans les valeurs des attributs de différentes classes, la séparabilité linéaire des données et la séparabilité entre les différentes classes). Dans un deuxième temps, nous allons étudier dans la section 3.5.2 la capacité des prétraitements à construire de bonnes matrices de Gram relativement à la variable cible. Finalement et avant de conclure, nous allons présenter dans la section 3.5.3 un exemple qui illustre la facilité d'interprétation des résultats de l'algorithme des K-moyennes quand il est précédé par le prétraitement supervisé Conditional Info.

3.5.1 La complexité des données

La complexité des problèmes de la classification supervisée est attribuée à plusieurs sources [18]. Parmi ces sources, on trouve l'ambiguïté des classes. Cette dernière fait référence à la situation où les instances de différentes classes ne peuvent pas être distinguées. Ceci pourrait être du à l'incapacité des variables de départ à décrire le concept cible : il se peut que ces variables ne sont pas très discriminantes pour le problème de la classification supervisée. Ce type de complexité ne peut être résolu au niveau du modèle d'apprentissage. Dans ce cas, un prétraitement des données s'avère nécessaire pour désambiguïser les classes. Un bon prétraitement est donc celui qui permet une description plus pertinente du concept cible.

Dans cette section, on cherche à comparer le comportement des différentes méthodes de prétraitement (supervisées et supervisées) présentées ci-dessus vis-à-vis d'un problème de classification supervisée. Il s'agit ici d'évaluer la capacité de chaque méthode de prétraitement à mieux décrire le concept cible par rapport à la description de départ. Une meilleure description permettra de facilité la tâche à l'algorithme des K-moyennes standard dans le but d'atteindre l'objectif des algorithmes de la classification supervisée. Par conséquent, une meilleure performances prédictive sera introduite. Cette problématique est connue sous le nom de la complexité des données.

Il existe dans la littérature des mesures permettant d'évaluer la complexité des données après un prétraitement des données. Ces mesures sont divisées en trois catégories. La première catégorie mesure le degré de chevauchement dans les valeurs des variables de différentes classes. La deuxième catégorie estime dans quelle mesure les classes sont séparables en examinant la longueur et la linéarité de la frontière de décision. La troisième catégorie mesure la compacité des groupes des différentes classes. Dans cette partie, nous allons utiliser la même mesure utilisée dans l'article [82], à savoir, les mesures F1, L2, N1 et N3. Pour plus de détails sur ces mesures, voir [81].

• **Le ratio discriminant maximal de Fisher (F1)** mesure la puissance discriminative maximale de chaque variable. La formule mathématique de F1 est donnée comme suit :

$$F1 = max_{j=1}^{d} FDR_j \tag{3.15}$$

Quand le problème de la classification est binaire (2 classes), le ratio pour chaque variable est alors calculé de la façon suivant :

$$FDR_j = \frac{(\mu_1^j - \mu_2^j)^2}{(\sigma_1^j)^2 + (\sigma_2^j)^2} \tag{3.16}$$

La mesure F1 est compris entre 0, et μ_k, avec μ_k est la moyenne de la variable j pour la classe k ($k \in \{1, 2\}$). Une valeur élevée de F1 signifie qu'au moins l'une des variables permet à l'algorithme d'apprentissage de séparer les instances de classes différentes en des partitions parallèles à un axe de l'espace des variables. Cependant, une faible valeur de F1 ne signifie pas que les classes ne sont pas linéairement séparables, mais plutôt qu'elles ne peuvent pas être discriminées par des hyperplans parallèles à l'un des axes de l'espace des variables.

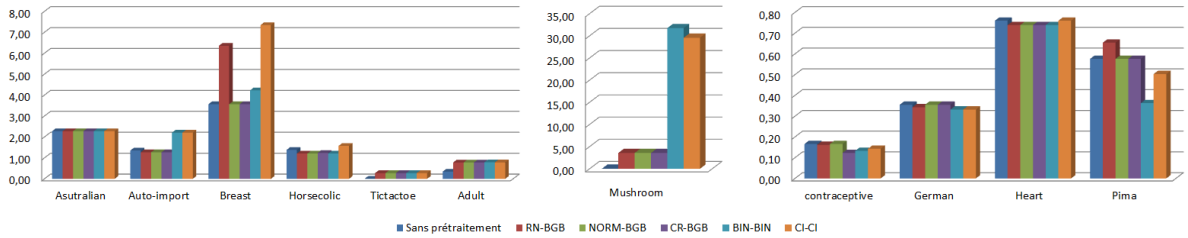


FIGURE 3.14 – La mesure F1 (à maximiser)

La figure 3.14 présente les performances des différentes méthodes de prétraitement en utilisant la mesure F1. Les résultats de cette figure montrent que les prétraitements supervisés CI-CI et BIN-BIN sont meilleurs ou très compétitifs avec les méthodes de prétraitement non supervisées. Ceci signifie que les méthodes de prétraitement supervisées parviennent à construire des variables synthétiques très discriminante pour le problème de la classification supervisée par rapport aux méthodes de prétraitement non supervisées et aux données de départ.

• **L'erreur d'apprentissage d'un classificateur linéaire (L2)** mesure le degré de la linéarité dans les données d'apprentissage. La mesure L2 commence par apprendre un algorithme d'apprentissage linéaire. Dans ce cas, l'algorithme des machines à vecteurs de support (SVM) [Vapnik, 1995] ayant un noyau linéaire et intégrant l'algorithme SMO [Platt, 1999] (Optimisation séquentielle minimale) est utilisé. Ensuite, la mesure renvoie l'erreur d'apprentissage sous forme de pourcentage des instances mal-classées. La mesure L2 est comprise entre 0 et 1. Une faible valeur de L2 signifie que les deux classes sont bien séparées.

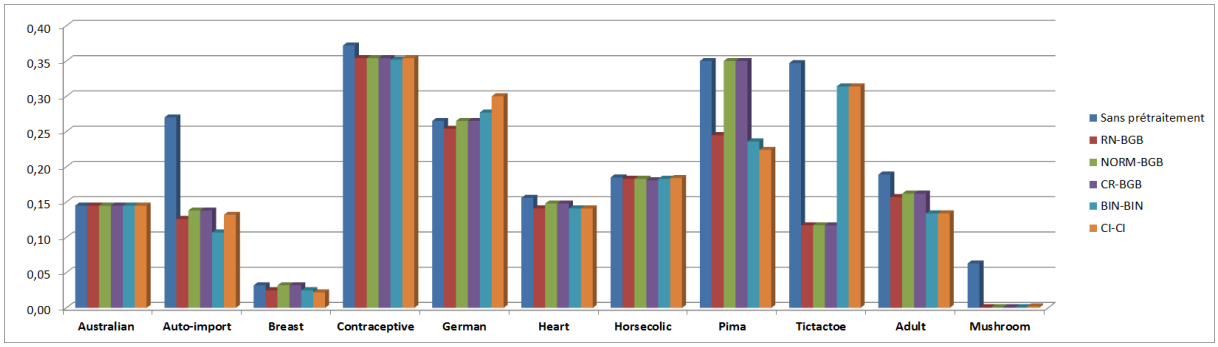


FIGURE 3.15 – La mesure L2 (à minimiser)

La figure 3.15 présente les performances des différentes méthodes de prétraitement en utilisant la mesure L2. Les résultats de cette figure montrent que les prétraitements supervisés CI-CI et

BIN-BIN sont meilleurs ou très compétitifs avec les méthodes de prétraitement non supervisées. Ceci signifie que les méthode de prétraitement supervisées parviennent à construire des données dont le degré de la linéarité est dans la plupart des temps, meilleur que celui des données du départ et des données générées par les méthodes de prétraitement non supervisées.

• **La proportion des points sur la frontière des classes (N1)** estime la longueur de la frontière des classes. Cette mesure est inspirée par le test proposé par Friedman et Rafsky [1979]. Elle commence par construire un arbre de recouvrement minimal (MST) sur l'ensemble des données en connectant tout d'abord tous les points à l'aide de la distance Euclidienne. Ensuite, elle retourne le rapport entre le nombre des nœuds de l'arbre de recouvrement qui relient les instances de différentes classes et le nombre total des instances dans l'ensemble de données. La mesure N1 est comprise entre 0 et 1. Une valeur élevée de cette mesure indique la majorité des points sont situés près de la frontière. Ceci peut rendre la tâche d'apprendre la frontière avec une bonne précision très difficile pour l'algorithme d'apprentissage.

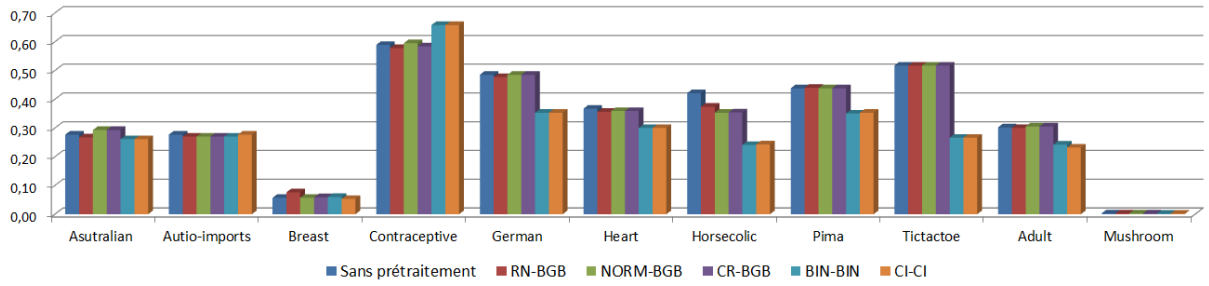


FIGURE 3.16 – La mesure N1 (à minimiser)

La figure 3.16 présente les performances des différentes méthodes de prétraitement en utilisant la mesure N1. Les résultats de cette figure montrent que les prétraitements supervisés CI-CI et BIN-BIN sont dans la plupart des temps, meilleurs que les méthodes de prétraitement non supervisées ((8/11) succès, (2/11) égalités et (1/11) échec). Ceci signifie que les méthodes de prétraitement supervisé parviennent à construire des données dont les classes sont bien séparées par rapport aux données du départ et aux données générées par les méthodes de prétraitement non supervisées.

• **Le taux d'erreur leave-one-out du classifieur '$1 - ppv$' (N3)** indique à quel point les instances de classes différentes sont proches. Elle renvoie le taux d'erreur de la validation leave-one-out pour le prédicteur K-plus proches voisins, avec K est fixé à 1. La mesure N3 varie dans l'intervalle $[0, 1]$. Les faibles valeurs de cette mesure indiquent qu'il existe un écart important dans la frontière des classes.

La figure 3.17 présente les performances des différentes méthodes de prétraitement en utilisant la mesure N3. Les résultats de cette figure montrent que les prétraitements supervisés CI-CI et BIN-BIN sont quasiment meilleurs que les méthodes de prétraitement non supervisées ((7/11) succès, (1/11) égalité et (3/11) échecs). Ceci signifie que les méthodes de prétraitement supervisé parviennent à construire des données dont les classes sont bien séparées par rapport aux données du départ et aux données générées par les prétraitements non supervisés.

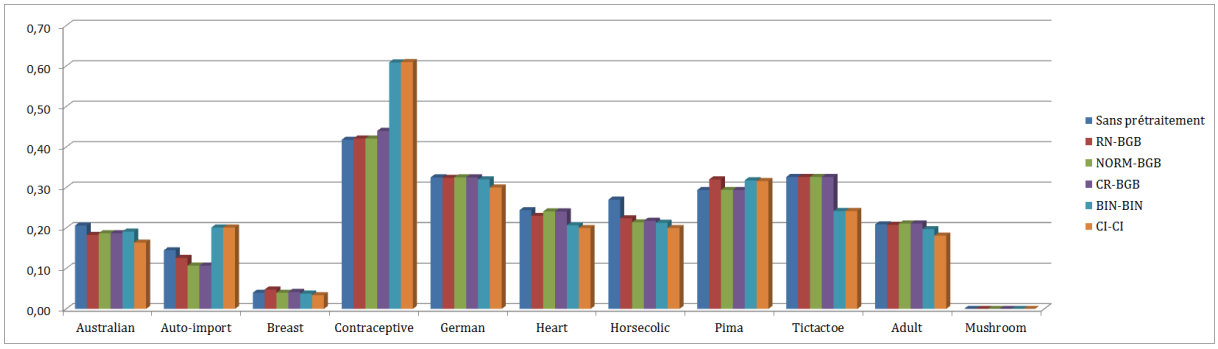


FIGURE 3.17 – La mesure N3 (à minimiser)

Discussion : Cette première étude expérimentale montre que nos méthodes de prétraitement parviennent dans la plupart des temps à construire des variables synthétiques discriminantes pour le problème de la classification supervisée par rapport aux variables de départ. Ensuite, les résultats expérimentaux montrent que nos méthodes de prétraitement supervisées des données parviennent dans la plupart des temps à générées des données dont les classes sont bien séparées par rapport aux données de départ et aux données générées par les méthodes de prétraitement non supervisées.

3.5.2 La similarité

Au niveau de l'axe de description, la comparaison de la qualité des résultats issus de l'algorithme des K-moyennes précédé par les différentes méthodes de prétraitements est une question très difficile. En effet, Ces méthodes n'ont pas la même plage de variation ni le même nombre de variables. Ceci rend l'utilisation des critères internes proposés dans le cadre du clustering inutile pour ce type de comparaison : ces critères se basent essentiellement sur une mesure de similarité pour évaluer la ressemblance entre les paires d'instances.

Un moyen pour surmonter ce problème, et de pouvoir comparer les différentes méthodes de prétraitement est d'évaluer, pour chaque jeu de données, la capacité des méthodes à construire de bonnes matrices de Gram relativement à la variable cible. Une matrice de Gram est une mesure de similitude entre les instances relativement à la variable cible. Une bonne matrice de Gram est donc celle qui produit une description plus concise de l'étiquetage des instances. Suivant ce contexte, la meilleure méthode de prétraitement est celle qui permet de construire des matrices de Gram pertinentes. Ceci reflète la capacité de la méthode à produire une bonne description des données vis-à-vis de la variable cible.

Il existe dans la littérature, un critère d'évaluation nommé EVA [48] permettant d'avoir une indication sur la capacité de chaque méthode de prétraitement à bien construire de bonnes matrices de Gram. La mesure EVA est une méthode qui prend en entrée une matrice de Gram et une variable cible catégorielle et renvoie un gain de compression. EVA mesure le gain qu'une partition établissant un compromis entre le nombre de groupes et la répartition des étiquettes peut apporter par rapport à la partition ayant un seul groupe. Celui-ci quantifie la capacité de la matrice de Gram à produire une description concise de l'étiquetage des instances. Elle évalue en général l'intérêt d'une matrice de Gram relativement à un problème de classification supervisée. EVA est une mesure à maximiser qui prend ses valeurs entre 0 et 1. Une valeur proche de 1 indique que la mesure de similitude relativement à la variable cible est très pertinente.

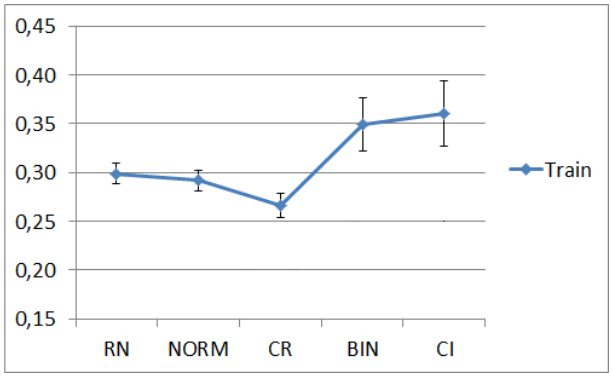


FIGURE 3.18 – La moyenne des performances en termes d'EVA sur 11 jeux de données (en test).

La figure 3.18 présente la moyenne des performances en termes d'EVA pour chaque méthode de prétraitement sur 11 jeux de données de l'UCI. Ces résultats sont obtenus en utilisant 100% des données en phase d'apprentissage. L'algorithme utilisé est l'algorithme des K-moyennes standard précédé par les différentes méthodes de prétraitement et la méthode d'initialisation des centres K-means++. Le nombre de clusters K considéré dans cette étude expérimentale est le nombre de classes J associé à chaque jeu de données.

Ces résultats montrent que la méthode de prétraitement Conditional Info (CI) suivie par la méthode Binarization (BIN) sont les deux méthodes de prétraitement qui produisent une meilleure description des données relativement à la variable cible et donc produisent des matrices de Gram pertinentes par rapport à celles produit par les méthodes de prétraitement non supervisées.

3.5.3 L'interprétation

L'interprétation des résultats issus de l'algorithme des K-moyennes prédictives est une condition incontournable dans cadre d'étude (voir la section 2.6.1 du chapitre 2). L'algorithme proposé dans cette thèse prend en entrée un biais de langage B permettant de bien décrire les données (voir l'algorithme générique du clustering prédictif (Algorithme 2) présenté dans la section 2.6.1 Chapitre 2). Ce biais de langage peut être vu par exemple comme des histogrammes permettant de connaître la répartition des variables dans chaque cluster appris.

Les méthodes de prétraitements proposées dans les deux sections 3.2.2 et 3.2.3 de ce chapitre sont en général des méthodes faciles à interpréter. En effet, la discrétisation supervisée des variables continues et le groupage supervisé en modalités des variables catégorielles rend l'interprétation "locale" des résultats issus de l'algorithme des K-moyennes plus facile. Cette interprétation locale permet à l'utilisateur de comprendre pour chaque cluster en particulier les facteurs les plus importants qui contribuent à sa construction. Pour illustrer la facilité d'interprétation des résultats issus de l'algorithme des K-moyennes standard précédé par les méthodes de prétraitement proposées (par exemple, Conditional Info), nous allons utiliser le jeu de données Adult de l'UCI. C'est un jeu de données caractérisé par la présence de 48842 instances, 15 variables descriptives et une variable cible possédant deux classes ("more", "less").

Pour ce cas illustratif, nous avons fixé le nombre de clusters à quatre. Les deux figures 3.19 et 3.20 dégagent quelques informations pertinentes concernant les instances du premier cluster de cette partition. Ces figures présentent pour chacune des variables (Relationship, Marital status, ..., etc) la proportion des instances appartenant à un groupe de modalité ou à un intervalle (selon la nature de la variable traitée) connaissant l'ensemble des données.

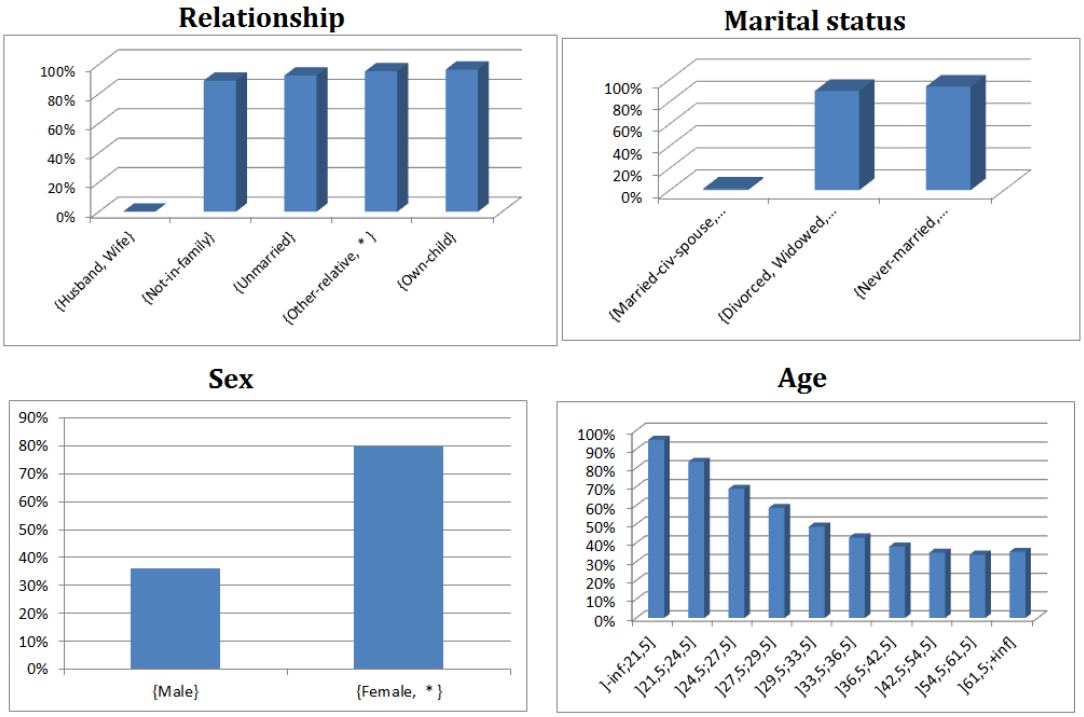


FIGURE 3.19 – L'interprétation locale d'un cluster d'une partition issus de l'algorithme des K-moyennes standard précédé par le prétraitement Conditional Info

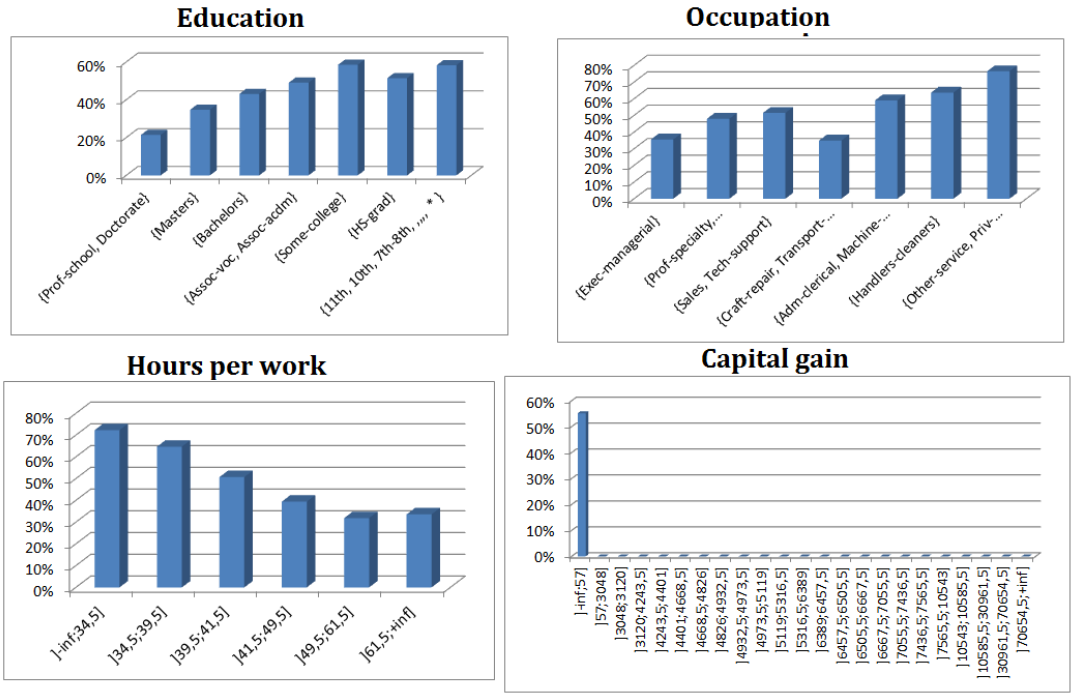


FIGURE 3.20 – L'interprétation locale d'un cluster d'une partition issue de l'algorithme des K-moyennes standard précédé par le prétraitement Conditional Info (Suite)

D'après les graphiques présentés dans la figure 3.19, On constate que :
— Pour la variable **Relationship** : le cluster ne contient pas de gens mariés. Par contre, il possède 92% des gens non mariés, 95% des gens ayant d'autres relations et 89% de gens sans famille existant dans de la base Adult.
— Pour la variable **Marital status** : les gens qui forment le cluster en question sont notamment soit divorcés, veufs ou leurs conjoints sont absents (90% de l'ensemble des données) ou bien séparés ou jamais mariés (94% de l'ensemble des données).
— Pour la variable **Sex** : Le cluster est formé des hommes et des femmes. Cependant, 80% des femmes existant dans l'ensemble des données Adult se trouvent dans ce cluster et seulement 36% des hommes.
— Pour la variable **Age** : le cluster contient tous les âges. L'information la plus importante ici est que 95% des gens de l'ensemble de données Adult ayant moins de 21.5 ans forment Ce cluster.

D'après les graphiques présentés dans la figure 3.20, On constate que :
— Pour la variable **Education** : Les gens de ce cluster ont des niveaux d'études variés. Par exemple, 95% des lycéens existant dans l'ensemble des données se trouvent dans ce cluster.
— Pour la variable **Occupation** : Les gens de ce cluster occupent des postes différents. À titre d'exemple, 51% des gens de l'ensemble des données travaillant dans le domaine de la vente se trouvent dans ce cluster.
— Pour la variable **Hours per work** : Puisque ce cluster possède des gens qui travaillent dans des postes divers, les heures de travail pendant la semaine varient également. Par exemple, 72% des gens travaillant moins de 34.5 heures par semaine existant dans l'ensemble de données se trouvent dans ce cluster.
— Pour la variable **Capital gain** : Ce cluster ne contient que des gens qui ont un capital gain inférieur à 57.

3.6 Bilan et synthèse

Ce chapitre a présenté l'influence d'une étape de prétraitement supervisé sur la qualité des résultats (au sens du clustering prédictif) générés par l'algorithme classique des K-moyennes. Tout d'abord, nous avons pu montrer que l'utilisation d'une distance dépendante de la classe construite à l'aide d'un prétraitement supervisé a la capacité d'aider l'algorithme des K-moyennes à atteindre l'objectif de clustering prédictif (du premier type) comparé aux méthodes non supervisées de prétraitement. En se basant sur l'ensemble des résultats obtenus dans la partie expérimentale, nous constatons que :

— *Pour l'axe de prédiction :* L'algorithme des K-moyennes standard précédé par le prétraitement supervisé Conditional Info parvient à fournir de bonnes performances prédictives en termes d'ARI par rapport aux performances obtenues lorsque l'algorithme est précédé par les méthodes non supervisées de prétraitement. La figure 3.21 présente la performance prédictive en termes d'ALC-ARI (en moyenne) obtenue sur 21 jeux de données lorsque l'algorithme des K-moyennes standard est précédé par Conditional Info et Rank Normalisation et/ou Basic Grouping . Cette figure confirme la conclusion tirée ci-dessus.

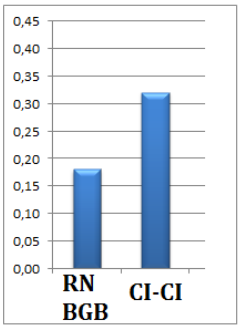


FIGURE 3.21 – la performance prédictive en termes d'ALC-ARI (en moyenne) obtenue sur 21 jeux de données lorsque l'algorithme des K-moyennes est précédé par CI-CI ou par RN-BGB

À travers l'utilisation d'un certain nombre de mesures de complexité des données, nous avons également pu montrer que nos méthodes de prétraitement supervisées des données (Conditional Info et Binarization) parviennent à construire dans la majorité des temps des variables synthétiques très discriminantes pour le problème de la classification supervisée. Les données générées par ces méthodes sont en général des données caractérisées par la présence des classes bien séparées les unes des autres par rapport aux données générées par les prétraitements non supervisées ou aux données de départ.

— *Pour l'axe de description :* Puisque les méthodes de prétraitements n'ont pas le même nombre de variables ni la même plage de variation, nous avons évalué l'axe de description en utilisant un critère permettant de mesurer la capacité des méthodes de prétraitements à construire de bonnes matrices de Gram relativement à la variable cible. Suivant ce contexte, nous avons pu montrer que le prétraitement supervisé Conditional Info suivi par le prétraitement Binarization parviennent à fournir de bonnes matrice de Gram par rapport aux prétraitements non supervisés.

— *Pour l'axe d'interprétation :* Dans la partie expérimentale, nous avons pu vérifier que la discrétisation supervisée des variables continues et le groupage en modalités des variables catégorielles permettent une interprétation aisée et plus concise des résultats issus de l'algorithme des K-moyennes.

Ensuite, nous avons pu montrer qu'avec la modification d'une seule étape de l'algorithme classique des K-moyennes (prétraitement des données), nous avons pu être compétitif (en termes de prédiction) face aux deux algorithmes de clustering supervisé les plus répandus dans la littérature et avec les arbres de décision tout en gardant une complexité algorithmique intéressante.

L'étape qui suit le prétraitement des données et qui semble avoir un impact direct sur la qualité des résultats issus des K-moyennes est l'étape d'initialisation des centres. Dans le cas de déséquilibre des classes à prédire (existence d'une classe majoritaire et d'une classe minoritaire), l'utilisation d'une méthode d'initialisation des centres non supervisée semble inappropriée. Par exemple, dans le cas où le nombre de clusters est égal au nombre de classes, la probabilité d'avoir plus d'un centre dans la classe majoritaire et de n'avoir aucun centre dans la classe minoritaire est élevée. Par conséquent, une détérioration au niveau de la prédiction va se produire. De ce fait, l'utilisation d'une étape d'initialisation supervisée des centres pourrait augmenter la performance de l'algorithme des K-moyennes en termes de prédiction. Cette étape d'initialisation fera donc l'objet du chapitre suivant.

Chapitre 4

Initialisation des centres

Sommaire

Ce chapitre a fait l'objet des publications suivantes :

[11] Oumaima Alaoui Ismaili, Vincent Lemaire, and Antoine Cornuéjols. Une initialisation des K-moyennes à l'aide d'une décomposition supervisée des classes. *Congrès de la Société Française de Classification (SFC)*, Nantes, 2015.

[60] Oumaima Alaoui Ismaili, Vincent Lemaire, and Antoine Cornuéjols. Une méthode supervisée pour initialiser les centres des k-moyennes. *Extraction et Gestion des Connaissances (EGC), Reims*, 2016.

[70] Vincent Lemaire, Oumaima Alaoui Ismaili, and Antoine Cornuéjols. An initialization scheme for supervized k-means. In *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland*, July 12-17, 2015, pages 1–8, 2015.

4.1 Introduction

L'algorithme des K-moyennes est l'un des algorithmes de clustering le plus répandu dans la littérature. Il doit sa popularité essentiellement à sa rapidité et à sa simplicité [61]. Cet algorithme consiste à construire une partition initiale des données de cardinalité K. A partir de cette étape d'initialisation, l'algorithme des K-moyennes cherche à améliorer itérativement le partitionnement en déplaçant les objets d'un groupe à un autre jusqu'à atteindre un critère terminal de stabilité. Pour plus de détail sur cet algorithme, le lecteur pourra se référer à [72, 61].

Afin d'atteindre notre objectif décrit dans la section 2.6.1 du chapitre 2, à savoir "la recherche d'un algorithme permettant de prédire et de décrire d'une manière simultanée", nous avons commencé par définir dans le chapitre 3, un prétraitement supervisé (interprétable) basé sur une estimation des distributions uni-variées conditionnellement aux classes. Ce prétraitement permet d'obtenir une distance dépendante de la classe qui aide l'algorithme des K-moyennes standard à avoir de bonnes performances au sens du clustering prédictif (i.e. le compromis entre la prédiction et la description). La deuxième étape qui suit l'étape de prétraitement est l'initialisation des centres (voir Algorithme 3 de la section 2.6.2 du Chapitre 2). Cette dernière a une grande influence sur les résultats fournis par l'algorithme des K-moyennes. Ce chapitre a donc pour but de discuter l'impact des méthodes d'initialisation supervisées sur la qualité des résultats de l'algorithme des K-moyennes classique au sens du clustering prédictif (ou la classification à base de clustering).

De par sa nature, l'algorithme standard des K-moyennes converge rarement vers un optimum global. La qualité [6] de cet optimum local et le temps requis par l'algorithme pour converger dépendent entre autre du choix des centres initiaux. Une mauvaise initialisation peut produire une solution (optimum local, «Figure 4.1 a») qui peut être très différente (ou loin) de la solution optimale «Figure 4.1 b». A titre d'exemple, la figure 4.1 présente une configuration dans laquelle l'algorithme des K-moyennes (K est fixé à 3) converge vers un minimum local qui ne reflète pas la vraie structure interne des données.

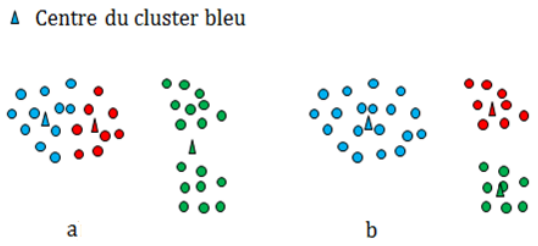


FIGURE 4.1 – Exemple de configuration dans lequel la solution générée par les K-moyennes (a) est très différente de la solution optimale (b)

Le choix d'une méthode d'initialisation appropriée est alors une étape très importante. L'utilisation d'une mauvaise méthode d'initialisation peut générer plusieurs effets indésirables tels que : *i*) des clusters vides, *ii*) une convergence plus lente, et *iii*) une grande probabilité de tomber sur un mauvais optimum local et donc la nécessité d'exécuter l'algorithme plusieurs fois [30]. Les centres choisis au départ doivent donc fournir une bonne couverture [7] de l'espace de données.

6. Une forte similarité à l'intérieur de chaque groupe et une forte dissimilarité entre les membres de différents groupes

7. Dans le but de minimiser la MSE, les centres initiaux doivent appartenir aux régions denses de l'espace d'entrée représentant des zones d'intérêt.

Ceci permet à l'algorithme d'obtenir un bon résultat sans avoir à l'exécuter de nombreuses fois, voire même en une seule fois si la méthode est déterministe.

Dans le cadre du clustering prédictif ou plus précisément des K-moyennes prédictives, le choix d'une méthode d'initialisation est également une problématique à résoudre. La seule différence réside dans le fait que l'utilisateur dispose dans ce cas d'une information supplémentaire à savoir : l'appartenance des instances à une des classes à prédire. Une solution optimale au sens des K-moyennes prédictives est donc celle qui réalise un bon compromis entre la compacité des groupes et leur pureté en termes de classes (pour plus de détails, voir Chapitre 2 Section 2.6).

Il est donc naturel de se demander si l'utilisation d'une méthode d'initialisation supervisée peut aider l'algorithme des K-moyennes standard à obtenir de bons résultats au sens du clustering prédictif. Une bonne méthode d'initialisation supervisée devrait réussir à capter les points appartenant aux régions denses et pures en termes de classes. L'obtention de ces points candidats peut faciliter la tâche de l'algorithme des K-moyennes standard puisqu'il va débuter avec une "bonne" partition initiale. Le but du travail présenté dans ce chapitre est donc de répondre à la question :

"À quel point une méthode d'initialisation supervisée pourrait aider l'algorithme des K-moyennes standard à former des groupes compacts, homogènes et purs au sens du clustering prédictif ? "

Pour répondre à cette question, nous allons tout d'abord présenté dans la section 4.2 un bref état de l'art des méthodes d'initialisation supervisées et non supervisées existantes dans la littérature. Ensuite, nous proposons dans la section 4.3 trois méthodes d'initialisation supervisées. Les résultats générés par l'algorithme des K-moyennes en utilisant ces méthodes supervisées et les méthodes issues de la littérature seront présentés et discutés dans les deux sections 4.5 et 4.6. Finalement, une conclusion générale contenant la réponse à la question posée ci-dessus sera présentée dans la section 4.7.

4.2 État de l'art

La recherche d'une méthode appropriée pour initialiser les centres des K-moyennes est un domaine de recherche très actif. Jusqu'à présent, une grande variété de méthodes existe dans la littérature. Dans cette section, nous allons présenter une brève description des méthodes les plus répandues. Ces méthodes peuvent être classées selon leur complexité en N (nombre d'instances du jeu de données) : linéaire, log-linéaire, quadratique, etc. Des articles de revue plus détaillés existent dans la littérature, le lecteur souhaitant une description plus détaillée de ces méthodes pourra se référer à [29], [28], [31], [66].

4.2.1 Les méthodes ayant une complexité linéaire en N

Les méthodes d'initialisation (quelle que soit leur complexité algorithmique), peuvent être catégorisées en deux grandes familles : déterministes et non déterministes. La première famille regroupe les méthodes capables de fournir un résultat unique quel que soit le nombre d'exécution de l'algorithme. Les résultats générés par l'algorithme des K-moyennes sont dans ce cas reproductibles. La deuxième famille, quant-à-elle, regroupe les méthodes basées sur l'aléatoire. Elles fournissent à chaque fois une solution différente de la précédente.

A. Les méthodes non déterministes

La première méthode d'initialisation proposée dans la littérature est celle de ***Forgy*** en 1965 [49]. Cette méthode consiste à choisir aléatoirement les K centres initiaux parmi les N instances de l'ensemble de données. La motivation derrière cette proposition réside dans le fait que la sélection aléatoire est susceptible de capter des points qui sont de bons candidats (par exemple, les points appartenant aux régions denses). Cependant, des points aberrants ou des points proches les uns des autres peuvent être choisis comme centres initiaux ce qui est clairement sous-optimal.

Bradley et Fayyad [24] ont proposé une méthode d'initialisation qui commence par une division aléatoire du jeu de données en B groupes. Ensuite, l'algorithme des K-moyennes est appliqué sur chacun de ces groupes en sélectionnant à chaque fois les centres aléatoirement. Les $B \times K$ centres obtenus sont alors regroupés et considérés comme une entrée de l'algorithme des K-moyennes. Ce dernier est ensuite exécuté B fois et est initialisé à chaque fois par des centres différents. Finalement, les centres qui forment la partition ayant une valeur minimale de MSE (*i.e.,* l'erreur quadratique moyenne) sont considérés comme les centres finaux. Le principal avantage de cette méthode est qu'elle augmente l'efficacité du résultat par le fait que les centres initiaux sont obtenus après des exécutions multiples de l'algorithme des K-moyennes. Cependant, l'inconvénient majeur de cette méthode est qu'elle nécessite beaucoup d'effort de calcul.

Sample [76] est une simple méthode d'initialisation qui consiste à appliquer un algorithme de partitionnement sur un échantillon de l'ensemble de données (souvent 10%). Les centres résultant sont alors considérés comme les centres initiaux. L'inconvénient majeur de cette méthode est qu'il se peut que l'échantillon sélectionné ne soit pas vraiment un échantillon représentatif de l'ensemble des données.

La méthode ***MaxiMin*** [52] choisie le premier centre aléatoirement. Puis le *ième* centre c_i ($i \in \{2, 3, \ldots, K\}$) est défini comme étant le point X_t qui vérifie :

$$t = argmax_{j \in \{1,2,\ldots,N\}}(min_{k \in \{1,2,\ldots,i-1\}} \parallel X_j - c_k \parallel_2^2)$$

Ce processus est répété $(K-1)$ fois.

La méthode ***K-means++*** [15] combine les deux méthodes ***Forgy*** et ***MaxiMin***. Cette méthode commence par choisir le premier centre aléatoirement. Ensuite le *i-ème* centre ($i \in \{2, \ldots, K\}$) est choisi de la manière suivante : *i*) calculer pour chaque point X' qui n'est pas un centre, la probabilité $\frac{dist(X')^2}{\sum_{X \in \mathcal{D}} dist(X)^2}$, où $dist(X)$ est la distance entre un point $X \in \mathcal{D}$ et son centroïde le plus proche, *ii*) tirer un centre c_i parmi les X' suivant cette probabilité, et *iii*) répéter les deux étapes *i*) et *ii*) jusqu'à ce que l'on ait placé tous les centres.

B. Les méthodes déterministes

MacQueen [72] a proposé une méthode simple d'initialisation des centres. Cette méthode consiste à prendre les K premiers points du jeu de données comme étant les centres initiaux. L'inconvénient majeur de cette méthode réside dans sa sensibilité envers l'ordre des données. De plus, les centres sélectionnés peuvent être proches les uns des autres.

Ball et Hall [17] proposent, quant à eux, une méthode d'initialisation qui consiste tout d'abord à choisir le centre de gravité de l'ensemble des données comme étant le premier centre. Ensuite, la distance entre ce centre et le premier point du jeu de données est alors calculée. Si cette distance est supérieure à un certain seuil T, alors ce point est choisi comme étant le deuxième centre. Sinon, le point suivant du jeu de données est alors testé. Ce processus est répété jusqu'à atteindre le nombre de clusters désiré. Le point fort de cette méthode est qu'elle

permet à l'utilisateur de contrôler la distance entre les centres de différents clusters. Cependant, ce procédé souffre de quelques inconvénients : *i*) la dépendance de la méthode à l'ordre des points dans le jeu de données, *ii*) la distance entre les centres dépend du seuil T qui doit être connu *a priori*. La complexité algorithmique de cette méthode est $\mathcal{O}(NKd)$.

La méthode de recherche des clusters simples [100] (*ou en anglais* ***Simple Cluster Seeking Method***) est similaire à la méthode proposée par Ball et Hall. La seule différence est dans le choix du premier centre. Cette méthode le considère comme étant le premier point dans le jeu de données. La complexité algorithmique de cette approche est $\mathcal{O}(NKd)$.

PCA-Part [97] utilise une approche de division hiérarchique basée sur l'analyse en composantes principales (ACP) pour déterminer les centres initiaux. La méthode part d'un seul cluster qui contient l'ensemble des données. Ensuite, elle sélectionne successivement le cluster ayant une valeur maximale de SSE (*i.e.,* Sum Squared Error) et le divise en deux sous-clusters à l'aide d'un hyperplan. Ce dernier passe par le centroïde du cluster sélectionné qui est orthogonal au vecteur propre principal de sa matrice de covariance. Cette procédure itérative est répétée $K-1$ fois. Finalement, les centres initiaux sont considérés comme étant les centroïdes des K groupes obtenus. Les différentes étapes de cette méthode sont comme suit :

1. Soit le cluster ayant la plus grande valeur de SSE [8] et μ est le centre de gravité du cluster C. Le premier cluster C_1 est le cluster contenant tous les données et c_1 est son centre de gravité.
2. Soit q la projection du c_i sur le principal vecteur propre $v_i (q = c_i.v_i)$
3. Diviser C_i en deux sous-clusters C_{i1} et C_{i2} en respectant la règle suivante :

$$\forall X_l \in C_i, \textbf{ Si } X_l.v_i \leq q \text{ alors } X_l \in C_{i1}$$
$$\textbf{Sinon } X_l \in C_{i2}$$

4. Répéter les étapes 1.-3.$(K-1)$ fois

Pour déterminer le vecteur propre principal à partir du cluster sélectionné, plusieurs méthodes peuvent être utilisées. A titre d'exemple, on trouve la méthode de la puissance [89] et la méthode Lanczos [22]. La complexité de cette méthode dépend alors de la méthode utilisée. Par exemple, la complexité de ***PCA-Part*** en utilisant la méthode puissance est $\mathcal{O}(Nd^2K)$.

Var-Part [97] est une approximation de la méthode PCA-Part. La seule différence entre les deux méthodes réside dans le choix de l'axe de projection. Dans Var-Part, à chaque itération, la matrice de covariance du cluster à diviser est supposée être diagonale. Dans ce cas, l'hyperplan de séparation est perpendiculaire à l'axe ayant la plus grande variance. Les différentes étapes de cette méthode sont :

- Soit C_1 le cluster contenant l'ensemble de données et c_1 le centre de gravité associé à ce cluster.

1. Sélectionner le cluster ayant la plus grande valeur de SSE.
2. Diviser le cluster C_i sélectionné en deux de la manière suivante :
   — Calculer la variance de chaque variable
   — Trouver la variable qui a une grande variance notée X^p avec $p \in \{1, \ldots, d\}$
   — Soit X_i^p la valeur de la variable X^p pour l'instance i et μ_i^p la moyenne du cluster C_j pour la variable p.
   Diviser le cluster C_j en deux sous-clusters C_{j1} et C_{j2} selon la règle suivante :

$$\textbf{Si } X_i^p \leq \mu_i^p \text{ alors } X_i \in C_{j1}$$
$$\textbf{Sinon } X_i \in C_{j2}$$

8. Soit pour un cluster C, $SSE = \sum_{X_j \in C} \parallel X_j - \mu \parallel_2^2$ avec $\parallel X_j \parallel_2 = (\sum_{l=1}^{d} X_{jl}^2)^{1/2}$

3. Répéter les deux étapes 1. et 2. $(K-1)$ fois.

La méthode ***KKZ*** est une méthode d'initialisation proposée par Katsavounidis et *al.* [64]. L'idée est de se focaliser sur les points les plus éloignés les uns des autres. Ces points sont les plus susceptibles d'appartenir à des clusters différents. La démarche suivie par la méthode KKZ est comme suit :

1. Choisir le point ayant la maximale norme ℓ_2 comme le premier centre.
2. Chaque centre c_j $(j \in \{2, \ldots, K\})$ est défini de la manière suivante : pour chaque point X_i du jeu de données qui n'est pas un centre, calculer la distance *dist* entre ce point et le centre le plus proche. Ensuite, le point ayant la plus grande valeur *dist* est choisi comme le centre c_j.

La méthode KKZ est connue par sa simplicité. Cependant, cette méthode est sensible à l'existence des outliers dans les données. La complexité algorithmique de cet algorithme est $\mathcal{O}(NKd)$.

4.2.2 Les méthodes ayant une complexité log-linéaire en N

La méthode de ***Hartigan*** [56] commence par trier les points du jeu de données en fonction de leurs distances au centre de gravité de l'ensemble de données. Le i*ème* centre $(i \in \{1, \ldots, K\})$, est défini comme étant le $(1 + (i-1)N/K)$*ème* point. Cette méthode est une amélioration de la méthode proposée par MacQueen [72]. Elle est invariante à l'ordre des données et semble générer des centres bien séparés.

La méthode ***ROBIN*** (ROBust INitialisation) [7] utilise le facteur local des outliers (LOF) [26] pour éviter la sélection des outliers comme des centres. A l'itération $i \in \{2, \ldots, K\}$, la méthode ROBIN trie les points du jeu de données dans un ordre décroissant en fonction de leur distance minimale aux centres déjà sélectionnés. Suivant ce tri ***ROBIN*** sélectionne le premier point rencontré ayant une valeur de LOF proche de 1 comme le i*ème* centre. Ce procédé est répété jusqu'à atteindre le nombre de clusters désiré.

Al-Daoud [5] trie tout d'abord les points en fonction de l'attribut ayant la plus grande variance et les partitionne ensuite en K groupes de même dimension. Les centres initiaux sont alors définis comme étant les points qui correspondent aux médianes de ces groupes. Cette méthode ne prend en compte qu'un seul attribut. Elle est susceptible d'être efficace seulement pour les données dont la variabilité est principalement contenue sur une seule dimension.

4.2.3 Les méthodes ayant une complexité quadratique en N

Kaufman et Rousseeuw [65] prennent le premier centre comme étant le centre de gravité de l'ensemble des données. Le i*ème* $i \in \{2, \ldots, K\}$ centre est choisi comme étant le point qui réduit le plus la valeur de SSE.

K. A. Abdul Nazeer *et al.* [78] proposent une méthode d'initialisation qui commence par calculer les distances entre chaque point de l'ensemble de données $\mathcal{D}$ et tous les autres points. Ensuite, le couple de points ayant la distance minimale est alors sélectionné et retiré du jeu de données en formant ainsi un nouveau sous-ensemble A_1. A chaque fois, le point le plus proche des points de ce sous-ensemble est alors ajouté à A_1 et retiré du jeu de données $\mathcal{D}$. Ce processus est répété jusqu'à ce que le nombre d'éléments dans A_1 atteint un certain seuil. De la même façon les sous-ensembles $A_2, \ldots, A_K$ sont construit à partir des points restant dans l'ensemble $\mathcal{D}$. Finalement, les centres initiaux sont obtenus en calculant les centres de gravité de chaque sous-ensemble $A_1, \ldots, A_K$.

Le tableau 4.1 présente quelques points résumant les méthodes d'initialisation qui seront utilisées dans les deux sections 4.5 et 4.6 lors des expérimentations.

Nom	**Type**	**Avantages et limites**	**Complexité**
Foggy (Random)	Non supervisée	- Simple et rapide - La probabilité de choisir des points proches ou des points aberrants est grande	$\mathcal{O}(K)$
Sample	Non supervisée	- Simple et rapide - L'échantillon sélectionné peut ne pas être un échantillon représentatif de l'ensemble de données	$\mathcal{O}(N'Kdt)$ N' est le nombre d'instances dans l'échantillon
Kmeans++	Non supervisée	- Génère des centres bien séparés les uns des autres - Deux sous-groupes proches peuvent partager le même centre même s'ils ont des classes différentes.	$\mathcal{O}(NKd)$
Var-Part	Non supervisée	- Déterministe, simple et rapide : sa complexité algorithmique est égale à la complexité des K-moyennes pour une seule itération - Elle se focalise sur la diminution de la MSE au sein de chaque cluster lors de la division sans prendre en considération qu'un groupe compact peut contenir des instances de classes différentes. Ceci dégrade la qualité au sens du clustering prédictif	$\mathcal{O}(NKd)$

TABLE 4.1 – Propriétés de quelques méthodes d'initialisation utilisées dans la partie d'expérimentation

4.3 Contribution

L'intérêt de l'utilisation d'une méthode supervisée pour initialiser les centres dans le cadre des K-moyennes prédictives peut être vu clairement dans le cas de déséquilibre des classes à prédire. Par exemple, dans l'exemple illustratif de la figure 4.2, si on tire aléatoirement les centres initiaux alors la probabilité de choisir plus d'un centre dans la classe majoritaire (*e.g.,* la classe rouge dans la figure 4.2) et de ne choisir aucun centre dans la classe minoritaire (*e.g.,* la classe magenta dans la figure 4.2) est élevée. Par conséquent, une détérioration au niveau de la pureté, en termes de classe à prédire, des clusters serait introduite. De ce fait, l'idée d'intégrer l'information contenue dans la variable cible dans le processus d'initialisation peut s'avérer nécessaire vis à vis du compromis entre description des données et prédiction des classes.

Cette section est consacrée à la présentation de trois nouvelles méthodes d'initialisation (K-means++R, Rocchio-and-split et S-Bisecting) qui se servent de l'information cible pour sélec-

tionner les centres initiaux.

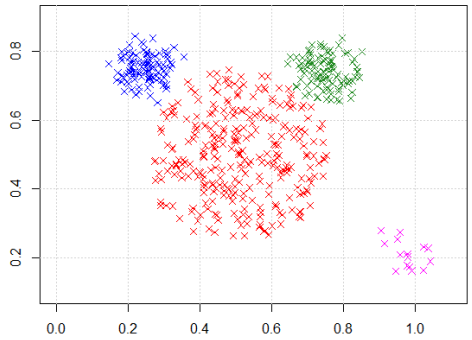


FIGURE 4.2 – Jeu de données Mouse [9][1]

4.3.1 K-means++R [70]

La première méthode proposée dans cette section est nommée K-means++R (ou K++**R**) [70]. Cette méthode suit un mécanisme dit : "d'*exploitation* et d'*exploration*". En effet, elle exploite dans un premier temps l'information donnée par la variable cible. Puis, elle explore dans un deuxième temps la densité de la distribution des données. Dans la première phase dite d'exploitation, K++R se sert de l'information donnée par la variable cible pour fournir les J premiers centres initiaux. Ces derniers sont obtenus de telle sorte que chacune des J classes contient un seul centre (*i.e.,* l'approche Rocchio [73]). Cette phase d'exploitation permet de résoudre la problématique citée ci-dessus : la probabilité de ne choisir aucun centre dans les classes minoritaires est dans ce cas nulle (voir Figure 4.2). Chaque centre est défini comme étant le centre de gravité des instances de la même classe. Par exemple, le $j ème$ centre associé à la classe C_j ($j \in \{1, \ldots, J\}$) est donnée par l'équation (4.1).

$$c_j = \frac{1}{N_j} \sum_{i \in C_j} X_i \tag{4.1}$$

La phase d'exploration, quant à elle, est consacrée à la sélection des $K - J$ centres initiaux restants. Cette phase cherche à explorer la densité de la distribution des données afin de sélectionner les points candidats appartenant aux régions denses. Cette phase est donc une étape non supervisée qui consiste à sélectionner à chaque itération le point le plus éloigné des centres déjà choisis. Il s'agit de chercher à assurer plus de diversité en explorant l'ensemble de données. La méthode K++R utilise à ce stade l'algorithme d'initialisation communément utilisé à savoir K-means++ [15]. Cet algorithme est débuté par les J centres trouvés dans la première phase (*i.e.*, la phase d'exploitation de la variable cible). Il est à noter que dans notre cadre d'étude, on ne s'intéresse pas au cas où le nombre de clusters K est inférieur au nombre de classes J puisqu'on cherche à découvrir la distribution sous-jacente de chaque classe à prédire.

La figure 4.3 présente un exemple illustratif de l'emplacement des centres initiaux dans l'espace des données (Figure 4.2) en utilisant la méthode K++R pour différent nombres de clusters $K \in \{4, 5, 6, 7\}$. Cette figure montre que : *i*) lorsque $K = J = 4$ (partie gauche de la figure), chaque classe (majoritaire et minoritaire) contient un centre, ce qui résout la problématique de

9. Le jeu de données Mouse est caractérisé par la présence de 500 instances, 2 variables descriptives et une variable à prédire contenant 4 classes.

déséquilibre des classes, *ii*) plus le nombre de clusters augmente ($K \in \{5, 6, 7\}$), plus la méthode cherche à trouver les points les plus éloignés les uns des autres en explorant l'ensemble des données.

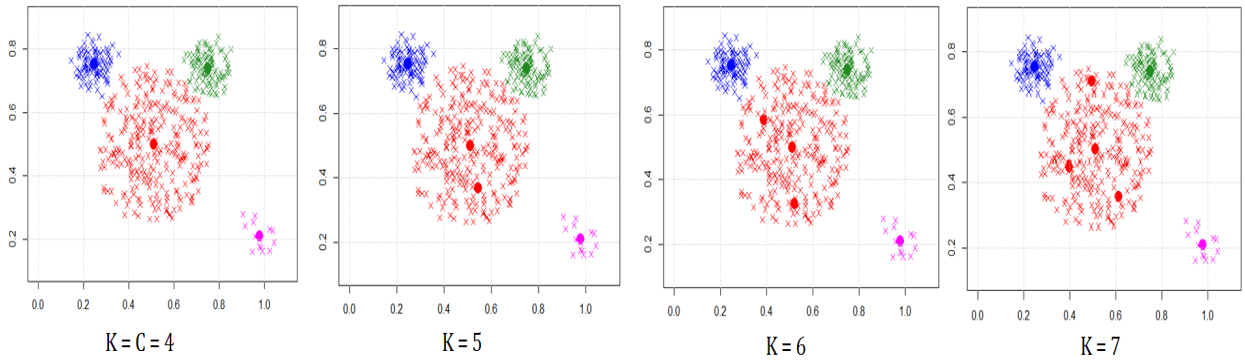


FIGURE 4.3 – l'emplacement des centres initiaux dans l'espace d'entrée en utilisant la méthode K-means++R pour $K \in \{4, 5, 6, 7\}$

Les différentes étapes de l'approche K++R sont présentées par des lignes de code présentées dans l'algorithme 4.

Entrée : $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{N}$ *: Le jeu de données d'apprentissage.*
K : Le nombre de clusters.
J : Le nombre de classes.

- Calculer le centre de gravité de chaque classe (équation 4.1)
Si $(J = K)$ **Alors**
  Sortie : les J centres de gravité
Fin Si
Si $(K > J)$ **Alors**
  On pose *dist* la distance entre un point X et son centroïde le plus proche
  1. Calculer pour chaque point X' qui n'est pas un centre la probabilité $\frac{dist(X')^2}{\sum_{X \in \mathcal{D}} dist(X)^2}$
  2. Tirer un centre c_i parmi les X' suivant cette probabilité
  3. Répéter 2. et 3. Jusqu'à ce que l'on ait placé tous les centres

  Sortie : Les K centres initiaux
Fin Si

Algorithme 4 – K-means++R

Dans le cadre du clustering standard, il est clair que cette méthode d'initialisation n'est pas une méthode appropriée pour former de bons clusters. En effet, le fait de dédier à chaque classe un centre peut générer une détérioration au niveau de la qualité des partitions générées en termes d'erreur de construction MSE (*ou* erreur quadratique moyenne). À titre d'exemple voir la figure 4.2 pour le cas où le nombre de clusters est égal au nombre de classes ($K = J = 4$). Cependant, dans le cadre du clustering prédictif, cette méthode est intéressante puisqu'on cherche plutôt à réaliser un compromis entre la compacité (*e.g.,* en termes de MSE) et la pureté en termes de

taux de bonnes classifications.

A note connaissance, il n'existe pas de méthode dans la littérature qui utilise une étape d'initialisation supervisée. Cependant, certains travaux dans le domaine semi-supervisé, intègrent l'information supplémentaire donnée par la variable cible pour initialiser les centres. La méthode d'initialisation proposée dans [19] dans le contexte semi-supervisé initialise de la même façon les centres initiaux que la méthode K++R. La principale différence réside dans l'étiquetage de la partition finale. Dans notre proposition, après la convergence de l'algorithme, on attribue à chaque cluster, la classe majoritaire correspondant aux instances qui le forme (i.e., le vote majoritaire). Par conséquent, l'étiquetage initial associé à la partition initiale change au cours du processus. Par contre dans [19], l'étiquetage des clusters reste inchangé : l'étiquetage initial et final est le même.

4.3.2 Méthodes basées sur la variance : Rocchio-And-Split et S-Bisecting [11, 60]

Lorsque le nombre de clusters (K) est supérieur au nombre de classes (J), la méthode K++R est partiellement supervisée. Les $K - J$ centres initiaux restants sont sélectionnés d'une manière non supervisée à l'aide de l'algorithme $(K - J)$Means++.Il est donc naturel de se demander si l'intégration de l'information contenue dans la variable cible lors de la sélection de ces $K - J$ centres initiaux restants pourrait garantir une meilleure performance que celle obtenue par la méthode K++R. Pour être en mesure de répondre à cette question, nous allons dans ce qui suit proposer deux nouvelles méthodes d'initialisation supervisées.

Ces deux méthodes d'initialisation sont appelées "Rocchio-And-Split" (RS) et "S-Bisecting" (SB). Dans le cas où le nombre de clusters (K) est égal au nombre de classes (J), ces deux méthodes fonctionnent de la même manière que K++R : elles dédient un seul centre à chaque classe. Chaque centre est définit comme étant le centre de gravité des instances de même classe (*i.e.,* la méthode Rocchio [73]). Dans le cas contraire, *i.e.,* lorsque le nombre de clusters est supérieur au nombre de classes, ces deux méthodes suivent une division hiérarchique descendante. Elles partent des J groupes où chaque groupe représente une classe. Ensuite, à chaque itération, le groupe qui vérifie un critère déterminé est alors divisé en deux. Ce processus est répété jusqu'à ce que le nombre de groupes formés soit égal au nombre de centres désiré. Au final, les centres initiaux sont obtenus en calculant les centres de gravité de chacun de ces groupes résultants. Les deux points clefs à déterminer sont alors : *comment sélectionner le groupe candidat à diviser ?*, et *comment le diviser ?*

- **Comment choisir le groupe candidat à diviser** ?

Dans la littérature, il existe plusieurs façons pour sélectionner le groupe candidat. Cette sélection dépend essentiellement du critère choisi et bien entendu du résultat attendu. Par exemple, on peut choisir de diviser le cluster i ayant la plus grande taille (*i.e.,* $i = argmin_k(1/n_k)$). Cette condition permet de produire des clusters de tailles équilibrées. On peut également choisir le groupe candidat suivant la similarité moyenne ou bien la cohésion [44]. Ces deux conditions permettent de produire des groupes compacts en se déplaçant d'un niveau de la hiérarchie à un autre (du haut vers le bas). Au lieu de se focaliser seulement sur les caractéristiques de chaque groupe, on peut également se baser sur la fonction objectif du clustering pour choisir le groupe candidat. Il s'agit de choisir le groupe k telle que la division de celui-ci conduit à une faible augmentation de la fonction objectif globale [44].

Dans notre cadre d'étude, le clustering prédictif cherche à «*discerner des groupes compacts,*

homogènes et purs en termes de classe». Les deux points essentiels qu'on peut retirer de cette définition sont la pureté et la compacité des clusters appris. La pureté peut être assurée en se basant sur le principe de la décomposition des classes (*e.g.,* [101]). C'est-à-dire, traiter chaque classe individuellement. En ce qui concerne, la compacité, elle peut être assurée en se basant sur les caractéristiques de chaque groupe. Il s'agit de chercher à diminuer la dispersion des données dans chaque groupe. En combinant les deux points, le but sera alors de diviser à chaque itération le groupe k_j (le groupe k ayant comme classe j) le plus dispersé. Ce processus est répété jusqu'à atteindre le nombre de groupes désiré. Dans notre proposition, on choisit de mesurer la dispersion par l'inertie intra-cluster.

- **Comment diviser le groupe candidat** ?

Après avoir sélectionné le groupe candidat, on cherche à le diviser en deux. Il est à noter que les deux méthodes proposées dans cette section (RS et SB) différent uniquement au niveau de la méthode de division du groupe candidat. Dans le reste de cette section, nous allons tout d'abord présenté l'approche Rocchio-And-Split tout en décrivant ses avantages et ses limites. Ensuite, nous allons présenter l'approche S-Bisecting.

A. L'approche "Rocchio-And-Split" (RS)

La méthode RS cherche à identifier les régions denses dans la classe ayant une variance intra-classe maximale. Pour ce faire, la méthode commence par sélectionner le groupe ayant une variance intra-classe élevée [10]. Ce groupe est considéré comme étant le groupe candidat à diviser. Pour le diviser, RS commence par sélectionner l'instance la plus éloignée du centre de gravité de ce groupe, notée $X_{i_{max}}$. Notons d_1 cette distance maximale et d_2 la distance entre $X_{i_{max}}$ et chaque instance du groupe à diviser. Ensuite, toutes les instances ayant une distance d_2 plus petite que la distance d_1 sont regroupées ensemble. Cela correspond tout simplement à diviser le cercle de rayon d_1 en deux. La figure 4.4 présente un exemple illustratif de la démarche de l'approche RS. Les différentes étapes de l'approche SB sont présentées par des lignes de code de l'algorithme 5.

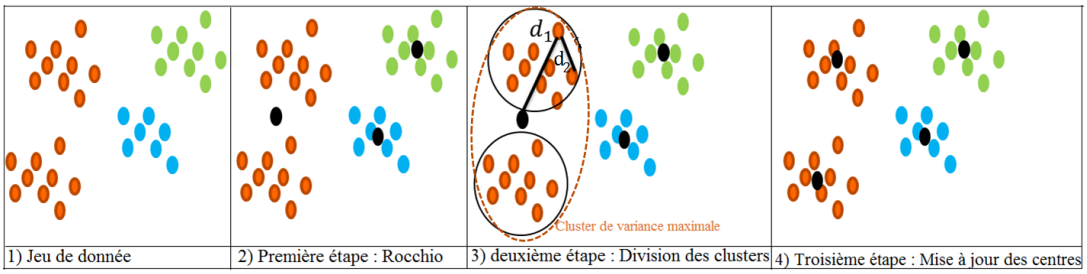


FIGURE 4.4 – Le fonctionnement de la méthode Rocchio-And-Split pour le cas où $K = 4$

10. Dans cette thèse, la variance intra-cluster est calculée à l'aide de l'inertie intra normalisée, présentée comme suit :

$$Intra(k) = \frac{1}{N_k} \sum_{X_i \in C_k} \| X_i - \mu_k \|^2$$

avec N_k est le nombre d'instances dans le cluster k ayant μ_k comme centre de gravité.

Entrée : $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{N}$ *: Le jeu de données d'apprentissage.*
K : Le nombre de clusters.
J : Le nombre de classes.

- Calculer le centre de gravité de chaque classe
Si $(J = K)$ **Alors**
  Sortie : les J centres de gravité
Fin Si
Si $(K > J)$ **Alors**
  Tant que (*le nombre de clusters n'est pas atteint*) **faire**
    - Calculer la dispersion dans chaque cluster en sens de l'inertie intra
    - Diviser le cluster le plus dispersé C_k en deux sous-clusters C_{k1} et C_{k2} de la manière suivante : Sélectionner l'instance $X_{i_{max}}$ de C_k telle que $i_{max} = argmax_{\{i \in C_k\}} dist(X_i, \mu_k)$ et $d_1 = dist(X_{i_{max}}, \mu_k)$ avec μ_k est le centre de gravité du cluster k
    Pour (j=1 ; j $\leq N_k$; j++) **faire**
      [N_k est le nombre d'instances dans C_k]
      Si $(d_1 \geq dist(X_{i_{max}}, X_j))$ **Alors**
        $X_j \in C_{k1}$
      Sinon
        $X_j \in C_{k2}$
      Fin Si
    Fin Pour
    - Supprimer le centre du groupe sélectionné et calculer les centres de gravité des deux clusters C_{k1} et C_{k2}
  Fait
  Sortie : Les K centres initiaux
Fin Si

Algorithme 5 – Rochio-And-Split

Avantages : L'un des points fort de l'approche RS est qu'elle est une approche déterministe. Cet avantage permet bien entendu de réduire le temps d'exécution. En effet, il n'est pas nécessaire d'exécuter l'algorithme des K-moyennes plusieurs fois afin de choisir le meilleur résultat (comme dans le cas des approches basées sur l'aléatoire). La complexité de cette approche est linéaire en N : $\mathcal{O}(CN_jd + KN_kd(K - J))$. De plus, la technique suivie pour diviser les groupes dispersés a pour but de capter les points appartenant aux régions denses et pures en termes de classe. L'approche RS est plus adaptée pour les données sphériques.

Limites : L'inconvénient majeur de cette approche est qu'elle est sensible à la présence des bruits, ou "outliers", en raison de l'utilisation de la distance maximale lors de la division. Néanmoins, cet inconvénient peut être atténué en utilisant par exemple un bon prétraitement des données (*e.g,* Conditional Info décrit dans la discussion de la Section 3.2.3 du Chapitre 3). Pour remédier à ce problème, l'approche S-Bisecting est donc proposée.

B. L'approche "S-Bisecting" (SB)

Pour l'approche S-Bisecting, la division du cluster le plus dispersé est réalisée en appliquant un algorithme des K-moyennes. Le nombre de clusters K est fixé ici à 2. La méthode d'initialisation utilisée au cours de la division est alors nécessairement une méthode non supervisée (car le cluster à diviser est pur). Pour S-Bisecting, on a choisi d'utiliser l'algorithme d'initialisation K-means++

(avec $K = 2$) [11] Les différentes étapes de l'approche SB sont présentées par des lignes de code de l'algorithme 6.

Entrée : $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{N}$ *: Le jeu de données d'apprentissage.*
K : Le nombre de clusters.
J : Le nombre de classes.

- Calculer le centre de gravité de chaque classe
Si $(J = K)$ **Alors**
  Sortie : les J centres de gravité
Fin Si
Si $(K > J)$ **Alors**
  Tant que (*le nombre de clusters n'est pas atteint*) **faire**
    - Calculer la dispersion dans chaque cluster en sens de l'inertie intra
    - Diviser le cluster le plus dispersé C_k en deux sous-clusters C_{k1} et C_{k2} de la manière suivante :
      1. Initialiser les centres à l'aide de 2-means++
      2. Appliquer l'algorithme 2-moyennes (initialisation K-means++) R fois
      3. Fixer les centres de gravité des deux clusters C_{k1} et C_{k2} à l'aide du résultat du meilleur des R 2-moyennes au sens de la SSE

  Fait
  Sortie : Les K centres initiaux
Fin Si

Algorithme 6 – S-Bisecting

Avantages : L'approche SB est une approche non déterministe. Elle contient une partie d'aléas lors de la division du cluster le plus dispersé. La complexité de la méthode SB est de l'ordre de $\mathcal{O}(R(CN_jd + 2N_kd(K - J)t))$ (t est le nombre d'itérations du 2-means et R est le nombre d'exécutions de l'algorithme (ou de réplicates)) qui est linéaire en N. Le principal avantage de cette approche est d'augmenter l'efficacité du résultat grâce au fait que les centres initiaux sont obtenus après des exécutions multiples de l'algorithme des K-moyennes ($K = 2$).

Limite : L'inconvénient majeur de cette méthode est qu'elle nécessite beaucoup d'efforts de calcul.

À notre connaissance, il n'existe pas dans la littérature de méthode d'initialisation supervisée pour les K-moyennes. Cependant, une méthode d'initialisation supervisée peut être construite en se basant sur le principe de la décomposition des classes (*e.g.,* [101]). Cette dernière consiste à attribuer un nombre égal de centres pour toutes les classes en utilisant l'algorithme K-means++. Le nombre de clusters doit être donc un multiplicateur du nombre de classes ($K = \beta \times J$). Par exemple, dans le cas où $K = (\beta \times J) + \alpha$ avec $(1 \leq \alpha < J)$, le nombre de clusters considéré restera $K = \beta \times J$. La méthode S-Bisecting (SB) proposée est plus parcimonieuse par rapport à la méthode de la décomposition de classes.

Si on revient maintenant à la question posée au début de ce chapitre, à savoir "*les méthodes d'initialisation supervisées peuvent-elles aider l'algorithme des K-moyennes standard à fournir de bons résultats au sens du clustering prédictif ?*". Pour vérifier sa validité, nous allons comparer les

11. Le choix de la méthode K-means++ n'est pas une obligation. D'autres méthodes d'initialisation peuvent également être utilisées (e.g., Variance Partitionning).

performances de l'algorithme des K-moyennes, précédé par différentes méthodes d'initialisation (supervisées ou non supervisées), en termes de compacité et de pureté. Le protocole expérimental suivi dans cette étude est présenté dans la section 4.4. Cette étude expérimentale est divisée en deux parties principales. La première partie est dédiée au cas où le nombre de clusters K est égal au nombre de classes J (Section4.5). Dans ce cas, on suppose que la variable à prédire ne dispose pas d'une structure interne à découvrir (*i.e.,* chaque classe à prédire est très compacte). Le problème est alors limité dans ce cas, au problème de la classification supervisée : on cherche à savoir si l'algorithme des K-moyennes, précédé par une étape d'initialisation supervisée, a la capacité de bien prédire la classe des nouvelles instances. La deuxième partie, quant à elle, est dédiée au cas où le nombre de clusters est supérieur au nombre de classes (Section 4.6). Dans ce cas, le problème devient plus complexe : on suppose que chaque classe (ou quelques-unes) a une structure qui la caractérise. Il s'agit donc de tester si l'algorithme des K-moyennes précédé par une méthode d'initialisation supervisée a la capacité de découvrir la structure interne de la variable cible.

4.4 Protocole expérimental

- Les méthodes d'initialisation : L'algorithme des K-moyennes doit sa popularité à l'une de ses propriétés, à savoir, sa rapidité : sa complexité est linéaire de l'ordre $\mathcal{O}(NKdt)$ (N : nombre d'instances, K : nombre de clusters, d : nombre de variables explicatives, t : nombre d'itérations). Pour préserver cet avantage, on s'intéresse aux méthodes d'initialisation ayant également une complexité linéaire en N [29]. Le tableau 4.2 présente l'ensemble des méthodes d'initialisation supervisées et non supervisées utilisé dans cette étude expérimentale.

Les méthodes non supervisées	**Les méthodes supervisées**
Forgy (**Random**)	K-means++R (**K++R**)
Sample (**Sample**)	Rocchio-and-Split (**RS**)
K-means++ (**K++**)	S-Bisecting (**SB**)
MaxiMin non déterministe (**MM(Rand)**)	Décomposition des classes (**CD**)
MaxiMin déterministe (**MM**)	
Var-Part (**Var-Part**)	

TABLE 4.2 – l'ensemble des méthodes d'initialisation utilisé

1. ***Les méthodes non supervisées*** : selon les études comparatives effectuées par Celebi et al. dans [31], *Var-Part* (Variance Partitioning) est l'une des méthodes qui fournit les meilleurs résultats. À côté de cette méthode, on utilise également les méthodes les plus répandues dans la littérature, à savoir : *K-means++* (K++), *MaxiMin* (déterministe (MM) et non déterministe (MM-Rand)), *Sample* et la méthode de *Forgy* (Random). Pour plus de détails sur ces méthodes, voir la section 4.2.
2. ***Les méthodes supervisées*** : En dehors de la méthode présentée dans [19], proche de K++R uniquement dans le cas où $K = J$ et qui diffère dans le processus d'étiquetage, il n'existe pas de méthodes d'initialisation supervisées. Nous décidons alors de prendre dans cette étude les méthodes proposées dans la section 4.3 (voir aussi la deuxième colonne du tableau 4.2). À côté de ces méthodes, nous avons ajouté dans la deuxième partie expérimentale, une autre méthode basée sur le principe de la décomposition des classes (CD). Pour plus de détails, voir la Section 4.3.2 b).

- Le prétraitement : dans cette étude expérimentale, nous avons choisi d'utiliser deux prétraitements. Le premier est un prétraitement supervisé nommé : "*Conditional Info*". Ce choix

fait suite à l'étude menée dans [10] et dans le chapitre 2 de ce mémoire où l'on a pu montrer que l'utilisation de ce prétraitement aide l'algorithme des K-moyennes standard à atteindre une bonne performance prédictive (le processus de prédiction est expliqué ci-dessous). Le deuxième prétraitement, quant à lui, est un prétraitement non supervisé. Parmi les prétraitements non supervisés, nous avons décidé d'utiliser celui qui fournit de bons résultats au sens du clustering prédictif (voir chapitre 3 section 3.4). Il s'agit de "*Rank Normalization*" (voir Chapitre 3 Section 3.3) pour les variables continues et de "*Basic-grouping*" (voir Chapitre 3 Section 3.3) pour les variables catégorielles. On prend ici deux types de prétraitements (supervisé et non supervisé) afin de savoir si la réponse à la question posée dans ce chapitre n'est pas dépendante du prétraitement utilisé.

- Nombre de clusters : Il varie de J jusqu'à K_i pour un prétraitement i utilisé. Pour chaque jeu de données, K_i a été déterminé au préalable de manière à ce que la partition obtenue, avec $K{=}K_i$ permette d'obtenir un ratio (inertie inter / inertie totale) de 80%. Pour plus de détails sur cette démarche voir l'annexe A. La valeur de K_i ($i \in \{1, 2\}$) pour chacune des jeux de données est indiquée dans le tableau 4.3 où $i = 1$ et $i = 2$ correspondent respectivement au Conditional Info et au Rank Normalization. Il est à noter que dans cette étude, le nombre de clusters K ne doit pas être inférieur à J puisqu'on suppose que la variable cible a une structure interne à découvrir.

ID	**Données**	M_n	M_c	N	J	K_1	K_2	J_{maj}
1	Iris	4	0	150	3	4	4	33
2	Hepatitis	6	13	155	2	9	66	79
3	Wine	13	0	178	3	12	38	40
4	Glass	10	0	214	6	15	25	36
5	Heart	10	3	270	2	23	90	56
6	Horsecolic	7	20	368	2	6	200	63
7	Soybean	0	35	376	19	20	49	14
8	Breast	9	0	683	2	4	12	65
9	Australian	14	0	690	2	22	210	56
10	Pima	8	0	768	2	10	74	65
11	Vehicle	18	0	846	4	11	24	26
12	Tictactoe	0	9	958	2	12	64	65
13	LED	7	0	1000	10	17	19	11
14	German	24	0	1000	2	7	363	70
15	Segmentation	19	0	2310	7	23	64	14
16	Abalone	7	1	4177	28	29	29	16
17	Waveform	21	0	5000	3	86	64	34
18	Adult	7	8	48842	2	12	64	76
19	Mushroom	0	22	8416	2	8	64	53
20	PenDigits	16	0	110992	10	64	64	10
21	Phoneme	256	0	2254	5	64	64	26

TABLE 4.3 – Liste des jeux de données utilisés - (J_{maj} représente $\approx$ pourcentage classe majoritaire)

- Les jeux de données : Pour évaluer et comparer les différentes méthodes d'initialisation en fonction de leur capacité à aider l'algorithme standard des K-moyennes à atteindre l'objectif du clustering prédictif, nous allons effectuer des tests sur différents jeux de données de l'UCI [1]. Ces jeux de données ont été choisis afin d'avoir des bases de données diverses en termes de nombre de classes J, de variables (continues M_n et/ou catégorielles M_c) et d'instances N (voir Tableau 4.3).

- Les critères d'évaluation : À notre connaissance, il n'existe pas dans la littérature un

critère global intégrant une partie interne qui mesure la compacité des clusters et une partie externe qui mesure la pureté des clusters en termes de classes. Pour cette raison, nous allons utiliser deux types de critères d'évaluation : supervisé (critère externe) et non supervisé (critère interne). Pour le critère supervisé, nous avons choisi un critère d'évaluation communément utilisé à savoir *Adjusted Rand Index* (ARI) [57]. En ce qui concerne le critère non supervisé, on a choisi d'utiliser *l'erreur quadratique moyenne* (MSE). La formule mathématique utilisée pour la MSE est donnée comme suit :

$$MSE = \frac{1}{N}\frac{1}{Z}\frac{1}{K}\sum_{i=1}^{N}\sum_{z=1}^{Z}\sum_{t=1}^{K}(XR_i^z - k_t^z)^2 \tag{4.2}$$

— N est le nombre d'instances dans l'ensemble de données.
— Z est le nombre de variable après le processus de prétraitement. Par exemple, pour Conditional Info, $Z = (M_n + M_c) \times J$.
— XR est le nouveau vecteur d'instance obtenu après le processus de prétraitement utilisé. Par exemple, pour Conditional Info, ce nouveau vecteur XR est de dimension $(M_n + M_c) \times J$.

- Nombre de réplicates (ou d'exécutions) R : De par sa nature, l'algorithme des K-moyennes standard converge rarement vers un optimum global. En utilisant une méthode d'initialisation basée sur l'aléatoire et en n'exécutant l'algorithme qu'une seule fois, ce dernier est susceptible de tomber sur un mauvais minima local. Afin d'éviter ce risque, ce dernier doit alors être exécuté plusieurs fois tout en changeant les conditions initiales. Dans cette étude, on prend $R \in \{1, 10, 100\}$.

- Le choix de la bonne partition : Lorsque l'algorithme des K-moyennes est exécuté R ($R > 1$) fois, la partition optimale (parmi les R partitions) est alors choisie suivant le critère MSE (voir équation 4.2 de la section 4.4).

- Affectation des classes aux clusters : À la fin du processus d'apprentissage, chaque groupe appris prend j comme étiquette si la majorité des exemples qui le forme sont de la classe j (*i.e.,* l'utilisation du vote majoritaire).

- La prédiction : à la présence d'une nouvelle instance, l'algorithme lui affecte l'étiquette du cluster qui lui est plus proche [12] (*i.e.,* l'utilisation du 1 plus proche voisin).

- Folds cross validation : pour pouvoir comparer les résultats obtenus, un 2×5 folds cross validation a été effectuée sur chaque jeu de données. De ce fait, les résultats sont présentés comme une moyenne de 10 tests.

4.5 Cas où le nombre de clusters (K) est égal au nombre de classes (J)

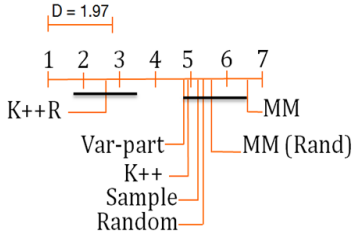
Dans le cas où le nombre de clusters est égal au nombre de classes ($K = J$), on suppose que la variable cible ne dispose pas d'une structure interne à découvrir (*i.e.,* chaque classe à prédire est compacte). Le problème du clustering prédictif devient tout simplement un problème de classification supervisée. Les groupes appris par l'algorithme doivent assurer une bonne performance en termes de pureté afin de pouvoir prédire correctement par la suite la classe des nouvelles instances. Dans cette étude, nous cherchons à savoir si les méthodes d'initialisation

12. Une instance i est plus proche au cluster C_1 que au cluster C_2 si et seulement $dist(i, g_1) < dist(i, g_2)$ avec g_1 (respectivement g_2) est le centre de gravité du cluster C_1 (respectivement C_2).

supervisées ont un impact sur les performances *'prédictives'* des K-moyennes. Il est à noter que les trois méthodes d'initialisation supervisées (*Rocchio-and-Split*, *S-Bisecting* et *K-means++R*) fonctionnent de la même façon lorsque $K = J$ (*i.e.,* l'utilisation du *Rocchio*). Pour cette raison, nous ne détaillons ci-dessous que les résultats de la méthode *K-means++R* (K++R) parmi ces trois méthodes.

Pour comparer les performances prédictives de plusieurs méthodes d'initialisation sur plusieurs bases de données, nous utilisons le test de Friedman couplé au test post-hoc de Nemenyi [41] pour un seuil de significativité $\alpha = 0.05$. Pour plus de détails sur ces deux tests voir Annexe B Section B.1.

• **Avec le prétraitement supervisé** : la figure 4.5 présente les résultats des comparaisons des performances prédictives en termes d'ARI de l'algorithme des K-moyennes en utilisant à chaque fois "Contidional Info" et l'une des méthodes d'initialisation. Ces résultats sont obtenus dans le cas où l'algorithme des K-moyennes n'est exécuté qu'une seule fois (*i.e.,* $R = 1$). Les méthodes dans la figure à gauche sont classées en ordre décroissant selon leurs performances prédictives en se basant sur la moyenne des rangs : plus le rang moyen de la méthode est proche de 1 meilleure elle est en prédiction. D'après le résultat du test de Friedman, il existe une différence significative entre les 7 méthodes d'initialisation ($p_{value} < 1.336e^{-07} \ll 0.05$) avec une grande préférence pour la méthode K++R. Ce résultat est confirmé par le test de Nemenyi (voir le tableau des p_{values} présenté dans la partie droite de la figure 4.5). Celui-ci partitionne les méthodes en deux groupes distincts $\{K++R\}$ et $\{Random, Var-Part, K++, Sample, MM(Rand), MM\}$ de tel sorte que le premier groupe (K++R) est celui qui fournit de bons résultats en termes de prédiction.



	Sample	Random	K++	Var-Part	MM (Rand)	MM
Random	1.000	-	-	-	-	-
K++	0.98935	0.99537	-	-	-	-
Var-Part	0.96807	0.98284	1.000	-	-	-
MM (Rand)	0.99996	1.000	0.99926	0.99537	-	-
MM	0.58212	0.50912	0.15911	0.10365	0.39190	-
K++R	0.00019	0.00031	0.00440	0.00845	0.00068	1.4e-08

FIGURE 4.5 – Comparaison des méthodes d'initialisation (précédées par le prétraitement "CI") en utilisant le test de Friedman couplé au test post-hoc de Nemenyi (ARI en test) pour R=1

Il faut noter que l'algorithme des K-moyennes converge très rarement vers un optimum global. Les méthodes d'initialisation basées sur l'aléatoire (*e.g.,* MM (Rand), Sample et Forgy) ont donc une grande chance d'échapper aux mauvais minima locaux lorsqu'on exécute l'algorithme de nombreuses fois. Pour cette raison, nous augmentons le nombre de réplicates $R \in \{10, 100\}$. Pour $R = 10$ (Figure 4.6 a)), le test de Friedman montre qu'il existe une différence significative ($p_{value} = 3.203e^{-09} \ll 0.05$) entre les méthodes, tandis que le test de Nemenyi partitionne les méthodes en deux groupes distincts : $\{K++R\}$ et $\{Sample, Random, Var-Part, MM, K++, MM(Rand)\}$. La figure (Figure 4.6 a)) montre que la méthode K++R est la meilleure en termes de prédiction. Pour $R = 100$ (Figure 4.6 b)), les mêmes conclusions peuvent être observées ($p_{value} = 1.526e^{-10}$ suivant le test de Friedman).

• **Avec le prétraitement non supervisé :** En utilisant Rank Normalization (RN) pour les variables continues et Basic Gouping (BGB) pour les variables catégorielles, la figure 4.7 présente les résultats des comparaisons des performances prédictives de l'algorithme des K-moyennes

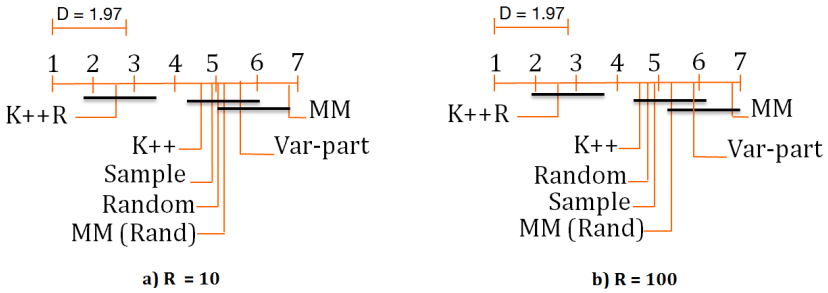


FIGURE 4.6 – Comparaison des méthodes d'initialisation (précédées par le prétraitement "CI") en utilisant le test de Friedman couplé au test post-hoc de Nemenyi (ARI en test) pour $R \in \{10, 100\}$

précédé à chaque fois par une méthode d'initialisation. Les résultats du test Friedman montrent que quel que soit le nombre de réplicates utilisé ($R \in \{1, 10, 100\}$), la méthode K++R reste celle qui fournit les meilleurs résultats en termes d'ARI. Tandis que le test de Nemenyi partitionne à chaque fois les méthodes en deux groupes. En appliquant le test bilatéral (Wilcoxon signé) pour les deux premières méthodes du classement (selon Friedman), on trouve que : 1) pour $R = 1$, $p_{value} = 10^{-4}$, 2) pour $R = 10$, $p_{value} = 0.007$ et 3) pour $R = 100$, $p_{value} = 10^{-4}$. On constate donc que quel que soit le nombre de réplicates utilisé, la méthode K++R est celle qui fournit les meilleurs résultats en termes de prédiction et elle est différente significativement de la deuxième méthode de classement.

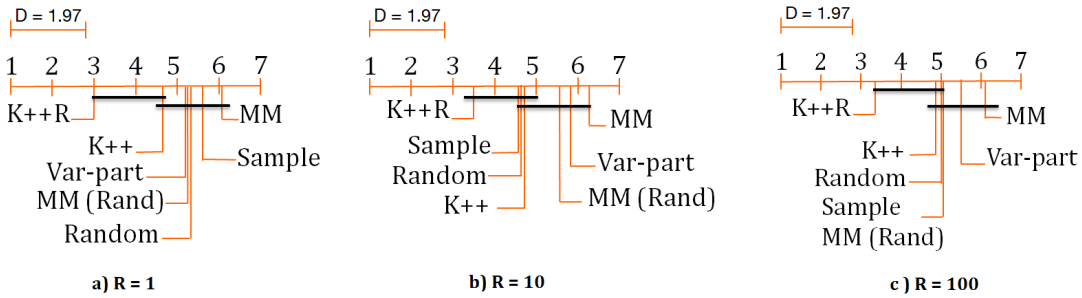


FIGURE 4.7 – Comparaison des méthodes d'initialisation (précédées par le prétraitement "RN" ou/et "Basic-Grouping") en utilisant le test de Friedman couplé au test post-hoc de Nemenyi (ARI en test)

Bilan

Selon les résultats des tests statistiques effectués dans cette première partie d'expérimentation, nous constatons que quel que soit le prétraitement utilisé, la méthode K++R fournit de meilleurs résultats en termes de prédiction par rapport aux autres méthodes. Le tableau 4.4 présente les résultats des performances prédictives moyennes (en termes d'ARI) en utilisant Conditional Info et Rank Normalization. Les résultats présentés pour la méthode K++R sont obtenus lorsque l'algorithme est exécuté une seule fois (*i.e.,* $R = 1$), tandis que les résultats présentés pour les autres méthodes sont obtenus lorsque l'algorithme est exécuté 100 fois (*i.e.,* $R = 100$). A partir de ces résultats, nous observons que la méthode K++R arrive à partir d'une seule exécution à fournir des résultats largement meilleurs pour quelques jeux de données ainsi que des résultats compétitifs pour le reste des jeux de données.

		R=100						R=1
	Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**K++R**
Conditional Info ($K = J$)	German	0.04 ± 0.04	0.04 ± 0.04	0.04 ± 0.04	0.03 ± 0.04	0.05 ± 0.04	0.07 ± 0.02	$\mathbf{0.12 \pm 0.02}$
	Australian	$\mathbf{0.5 \pm 0.06}$	$\mathbf{0.5 \pm 0.06}$	$\mathbf{0.5 \pm 0.06}$	0.49 ± 0.07	$\mathbf{0.5 \pm 0.06}$	0.35 ± 0.09	$\mathbf{0.5 \pm 0.06}$
	LED	0.47 ± 0.04	0.48 ± 0.03	0.48 ± 0.03	0.41 ± 0.04	0.44 ± 0.02	0.27 ± 0.02	$\mathbf{0.53 \pm 0.03}$
	Hepatitis	0.05 ± 0.06	0.05 ± 0.06	0.05 ± 0.06	0.03 ± 0.07	0.08 ± 0.1	0.27 ± 0.15	$\mathbf{0.20 \pm 0.11}$
	Heart	0.30 ± 0.11	0.30 ± 0.11	0.30 ± 0.11	0.17 ± 0.09	0.30 ± 0.11	0.22 ± 0.11	$\mathbf{0.36 \pm 0.09}$
	Glass	0.52 ± 0.12	0.52 ± 0.12	0.53 ± 0.11	0.79 ± 0.06	0.49 ± 0.09	0.30 ± 0.01	$\mathbf{0.82 \pm 0.06}$
	Breast	0.87 ± 0.06	0.87 ± 0.06	0.87 ± 0.06	0.87 ± 0.06	0.87 ± 0.06	0.8 ± 0.14	$\mathbf{0.88 \pm 0.04}$
	Iris	0.62 ± 0.08	0.62 ± 0.08	0.62 ± 0.08	0.61 ± 0.09	0.62 ± 0.08	0.60 ± 0.07	$\mathbf{0.72 \pm 0.07}$
	Pima	-0.02 ± 0.02	-0.02 ± 0.02	-0.02 ± 0.02	-0.02 ± 0.01	-0.02 ± 0.02	0.07 ± 0.08	$\mathbf{0.10 \pm 0.07}$
	Wine	0.91 ± 0.06	0.91 ± 0.06	0.91 ± 0.06	0.86 ± 0.06	0.86 ± 0.06	0.66 ± 0.14	$\mathbf{0.92 \pm 0.06}$
	Tictactoe	0.11 ± 0.03	0.11 ± 0.03	0.11 ± 0.03	0.11 ± 0.03	0.10 ± 0.02	0.05 ± 0.05	$\mathbf{0.14 \pm 0.02}$
	Vehicle	0.17 ± 0.02	0.17 ± 0.02	0.17 ± 0.02	$\mathbf{0.19 \pm 0.02}$	0.14 ± 0.02	0.1 ± 0.04	$\mathbf{0.19 \pm 0.04}$
	Horsecolic	$\mathbf{0.37 \pm 0.06}$	0.37 ± 0.06	$\mathbf{0.37 \pm 0.06}$	0.29 ± 0.1	0.14 ± 0.1	0.11 ± 0.07	$\mathbf{0.37 \pm 0.06}$
	Abalone	0.05 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	$\mathbf{0.06 \pm 0.01}$	$\mathbf{0.06 \pm 0.01}$	$\mathbf{0.06 \pm 0.01}$	$\mathbf{0.06 \pm 0.01}$
	Segmentation	$\mathbf{0.70 \pm 0.04}$	$\mathbf{0.70 \pm 0.04}$	$\mathbf{0.70 \pm 0.04}$	0.67 ± 0.03	0.7 ± 0.03	0.63 ± 0.03	0.68 ± 0.02
	Soybean	0.52 ± 0.05	0.53 ± 0.04	0.53 ± 0.06	0.45 ± 0.02	0.51 ± 0.04	0.42 ± 0.03	$\mathbf{0.61 \pm 0.04}$
	Waveform	0.22 ± 0.02	0.22 ± 0.02	0.22 ± 0.02	0.22 ± 0.02	0.22 ± 0.02	0.14 ± 0.07	$\mathbf{0.23 \pm 0.04}$
	Adult	0.15 ± 0.02	0.15 ± 0.02	0.15 ± 0.02	0.15 ± 0.02	0.15 ± 0.02	0.04 ± 0.07	$\mathbf{0.17 \pm 0.02}$
	Mushroom	$\mathbf{0.94 \pm 0.01}$	$\mathbf{0.94 \pm 0.01}$	$\mathbf{0.94 \pm 0.01}$	$\mathbf{0.94 \pm 0.01}$	$\mathbf{0.94 \pm 0.01}$	0.06 ± 0.01	$\mathbf{0.94 \pm 0.01}$
	PenDigits	0.56 ± 0.01	0.56 ± 0.02	0.56 ± 0.02	0.51 ± 0.02	0.56 ± 0.02	0.48 ± 0.02	$\mathbf{0.62 \pm 0.01}$
	Phoneme	$\mathbf{0.73 \pm 0.01}$	$\mathbf{0.73 \pm 0.01}$	$\mathbf{0.73 \pm 0.01}$	0.72 ± 0.01	$\mathbf{0.73 \pm 0.01}$	$\mathbf{0.73 \pm 0.01}$	$\mathbf{0.73 \pm 0.01}$

		R=100						R=1
	Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**K++R**
Rank Normalization + BGB ($K = J$)	German	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	$\mathbf{0.02 \pm 0.02}$	0.01 ± 0.01
	Australian	$\mathbf{0.22 \pm 0.02}$	$\mathbf{0.22 \pm 0.02}$	$\mathbf{0.22 \pm 0.02}$	$\mathbf{0.22 \pm 0.02}$	$\mathbf{0.22 \pm 0.02}$	0.04 ± 0.06	$\mathbf{0.22 \pm 0.02}$
	LED	0.47 ± 0.02	0.47 ± 0.02	0.48 ± 0.02	0.45 ± 0.01	0.47 ± 0.01	0.36 ± 0.02	$\mathbf{0.51 \pm 0.01}$
	Hepatitis	0.17 ± 0.11	0.17 ± 0.11	0.17 ± 0.11	0.17 ± 0.11	0.17 ± 0.11	-0.01 ± 0.03	$\mathbf{0.19 \pm 0.12}$
	Heart	$\mathbf{0.40 \pm 0.06}$	$\mathbf{0.40 \pm 0.06}$	$\mathbf{0.40 \pm 0.06}$	0.39 ± 0.05	$\mathbf{0.40 \pm 0.06}$	0.25 ± 0.14	$\mathbf{0.40 \pm 0.06}$
	Glass	0.29 ± 0.06	0.29 ± 0.05	$\mathbf{0.3 \pm 0.04}$	0.26 ± 0.05	0.24 ± 0.04	0.23 ± 0.04	$\mathbf{0.3 \pm 0.07}$
	Breast	$\mathbf{0.90 \pm 0.03}$	$\mathbf{0.90 \pm 0.03}$	$\mathbf{0.90 \pm 0.03}$	$\mathbf{0.90 \pm 0.03}$	$\mathbf{0.90 \pm 0.03}$	$\mathbf{0.90 \pm 0.03}$	$\mathbf{0.90 \pm 0.03}$
	Iris	$\mathbf{0.66 \pm 0.09}$	$\mathbf{0.66 \pm 0.09}$	$\mathbf{0.66 \pm 0.09}$	0.64 ± 0.08	$\mathbf{0.66 \pm 0.09}$	0.65 ± 0.09	0.64 ± 0.09
	Pima	0.08 ± 0.02	0.08 ± 0.02	0.08 ± 0.02	0.09 ± 0.02	0.08 ± 0.02	0.10 ± 0.03	$\mathbf{0.11 \pm 0.02}$
	Wine	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	0.87 ± 0.05	0.88 ± 0.06	0.87 ± 0.04	$\mathbf{0.9 \pm 0.06}$
	Tictactoe	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	$\mathbf{0.07 \pm 0.01}$	0.01 ± 0.01	0.01 ± 0.01	$\mathbf{0.07 \pm 0.02}$
	Vehicle	0.08 ± 0.01	0.08 ± 0.01	0.08 ± 0.01	0.08 ± 0.01	0.08 ± 0.01	0.08 ± 0.01	$\mathbf{0.12 \pm 0.01}$
	Horsecolic	0.08 ± 0.04	0.09 ± 0.05	0.09 ± 0.04	0.08 ± 0.04	0.08 ± 0.04	0.02 ± 0.02	$\mathbf{0.12 \pm 0.05}$
	Abalone	$\mathbf{0.05 \pm 0.0}$	$\mathbf{0.05 \pm 0.0}$	$\mathbf{0.05 \pm 0.01}$	$\mathbf{0.05 \pm 0.0}$	$\mathbf{0.05 \pm 0.0}$	$\mathbf{0.05 \pm 0.0}$	$\mathbf{0.05 \pm 0.01}$
	Segmentation	0.52 ± 0.01	0.52 ± 0.01	0.52 ± 0.01	0.50 ± 0.01	0.52 ± 0.01	0.52 ± 0.01	$\mathbf{0.54 \pm 0.01}$
	Soybean	0.52 ± 0.04	0.51 ± 0.05	0.5 ± 0.04	0.4 ± 0.03	0.52 ± 0.03	0.39 ± 0.02	$\mathbf{0.64 \pm 0.04}$
	Waveform	0.26 ± 0.0	0.26 ± 0.0	0.26 ± 0.0	0.26 ± 0.0	0.26 ± 0.0	0.26 ± 0.0	0.28 ± 0.04
	Adult	0.14 ± 0.05	0.14 ± 0.05	0.14 ± 0.05	0.14 ± 0.05	0.14 ± 0.05	0.14 ± 0.05	$\mathbf{0.18 \pm 0.01}$
	Mushroom	$\mathbf{0.62 \pm 0.01}$	$\mathbf{0.62 \pm 0.01}$	$\mathbf{0.62 \pm 0.01}$	$\mathbf{0.62 \pm 0.01}$	$\mathbf{0.62 \pm 0.01}$	0.01 ± 0.01	$\mathbf{0.62 \pm 0.0}$
	PenDigits	0.57 ± 0.02	0.57 ± 0.01	0.57 ± 0.01	0.59 ± 0.02	0.57 ± 0.01	0.57 ± 0.02	$\mathbf{0.63 \pm 0.04}$
	Phoneme	$\mathbf{0.61 \pm 0.01}$	$\mathbf{0.61 \pm 0.01}$	$\mathbf{0.61 \pm 0.01}$	$\mathbf{0.61 \pm 0.01}$	$\mathbf{0.61 \pm 0.01}$	0.46 ± 0.03	$\mathbf{0.61 \pm 0.01}$

TABLE 4.4 – Performance prédictive en terme d'ARI lorsque $K = J$ en utilisant conditional Info et Rank Normalization

4.6 Cas où le nombre de clusters (K) est supérieur au nombre de classes (J)

Lorsque le nombre de clusters est supérieur au nombre de classes, les approches de clustering prédictif cherchent à décrire et à prédire d'une manière simultanée. L'objectif est alors de trouver au cours de la phase d'apprentissage, le meilleur compromis entre la compacité et la pureté des groupes appris. Il s'agit de découvrir la structure interne de la variable cible. La prédiction de la classe des nouvelles instances est réalisée en se basant sur cette structure. Puisqu'il n'existe pas de critère global permettant de mesurer ce compromis, dans ce qui suit les performances prédictives des méthodes seront évaluées en utilisant l'ARI et la compacité des groupes formés sera évaluée en utilisant la MSE. Finalement, une discussion sera mener sur le compromis prédiction\compacité. L'ensemble des tableaux qui servent à obtenir les résultats synthétiques présentés dans cette deuxième partie d'expérimentation sont détaillés dans l'Annexe C.

4.6.1 Évaluation de la prédiction

Pour cet axe d'évaluation, quel que soit le prétraitement utilisé, nous commençons par tracer pour chaque jeu de données et pour chaque méthode d'initialisation la courbe d'ARI en fonction du nombre de clusters (de $K = J$ jusqu'à K_1 pour Conditional Info et de $K = J$ jusqu'à K_2 pour Rank Normalization). Pour plus de détails sur la manière d'obtention de K_1 et de K_2 , voir Section 4.4 et\ou l'Annexe A de ce mémoire. L'aire sous cette courbe est ensuite calculée (ALC-ARI : Area under the Learning Curve de l'ARI). Dans ce cas, plus la valeur de l'aire est grande plus la méthode est bonne. La figure 4.8 présente un exemple illustratif de l'aire sous la courbe d'ARI calculée pour la méthode SB (précédée par Conditional Info) pour le jeu de données Heart. Le nombre de clusters K varie dans ce cas de $J = 2$ jusqu'à $K_1 = 23$.

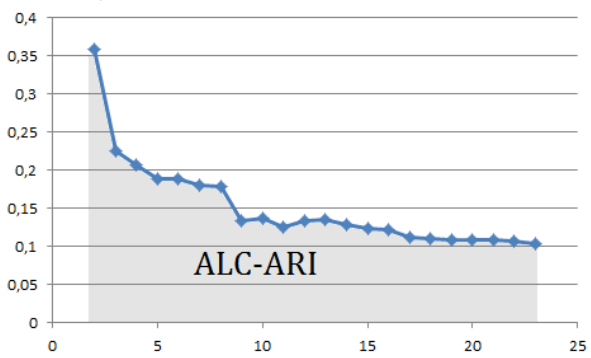


FIGURE 4.8 – L'aire sous la courbe d'ARI pour la méthode SB pour le jeu de données Heart en utilisant CI ($K_1 = 23$)

Muni des valeurs d'ALC-ARI, qui synthétisent les résultats de chaque méthode d'initialisation (voir tableau 4.2 de la section 4.4), nous appliquons le test de Friedman couplé au test post-hoc de Nemenyi sur les 21 jeux de données et pour un seuil de significativité $\alpha = 0.05$. Les valeurs d'aires qui ont servi à l'obtention de ces tests statistiques sont disponibles dans les tableaux C.4 et C.5 de l'Annexe C.

Avec le prétraitement supervisé : suivant les résultats du test statistique de Friedman, on observe que les méthodes d'initialisation sont différentes significativement quel que soit le nombre

de réplicates (R) utilisé. En outre, en s'appuyant sur les résultats du test de Nemenyi présentés dans la figure 4.9, on constate que ces méthodes peuvent être partitionnées en 4 groupes de telle sorte que la méthode RS suivie par les deux méthodes SB et CD sont celles qui fournissent de bons résultats en termes d'ARI. On observe également que lorsque le nombre de réplicates (R) augmente, les trois méthodes RS, SB et CD s'écartent des autres méthodes et deviennent plus performantes en termes de prédiction.

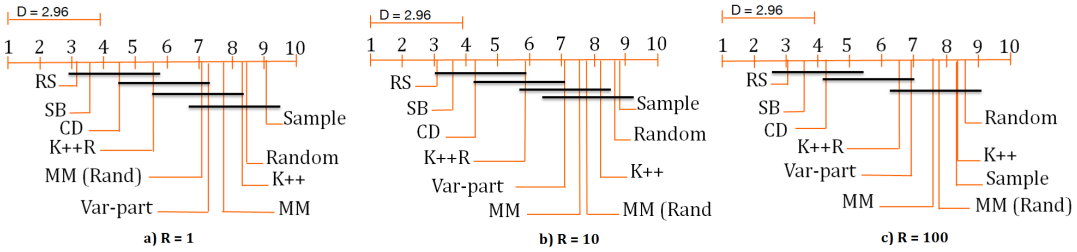


FIGURE 4.9 – Comparaison des méthodes d'initialisation (précédées par Conditional Info) en utilisant le test de Friedman couplé au test post-hoc de Nemenyi (ALC-ARI en test)

Avec le prétraitement non supervisé : même lorsqu'on change le prétraitement supervisé (CI) par le prétraitement non supervisé (RN et/ou BG), nous trouvons les mêmes résultats : la méthode RS suivie par les deux méthodes SB et CD sont les méthodes les plus performantes en termes d'ARI (voir la figure 4.10).

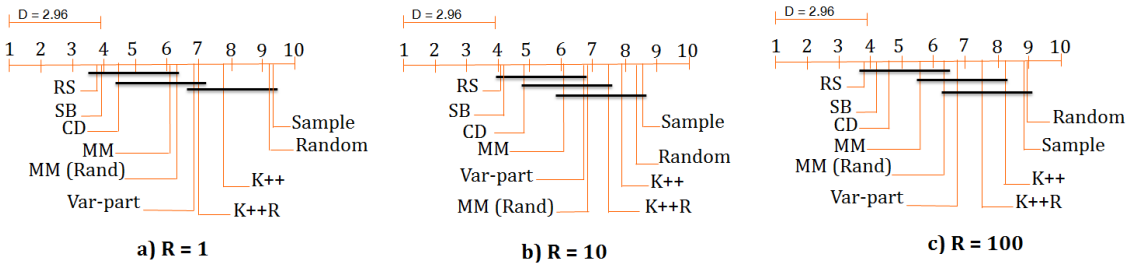


FIGURE 4.10 – Comparaison des méthodes d'initialisation (précédées par RN-BGB) en utilisant le test de Friedman couplé au test post-hoc de Nemenyi (ALC-ARI en test)

4.6.2 Évaluation de la compacité

Pour l'axe de la description, nous procédons de la même manière que pour la prédiction : pour chaque méthode d'initialisation et pour chaque jeu de données, nous traçons la courbe de l'erreur quadratique moyenne MSE (voir l'équation 4.2 de la Section 4.4) en fonction du nombre de clusters puis nous calculons l'aire sous la courbe (ALC-MSE). Dans ce cas, plus la valeur de l'aire est petite meilleure est la méthode. Les valeurs d'aires qui ont servi à l'obtention de ces tests statistiques sont disponibles dans les tableaux C.6 et C.7 de l'Annexe C.

Avec le prétraitement supervisé : Lorsque $R = 1$, on remarque que les quatre méthodes Var-Part, SB, K++ et K++R sont meilleures que les autres méthodes en termes de compacité des groupes appris (la figure 4.11 a)). Lorsqu'on augmente R (*i.e.,* $R \in \{10, 100\}$), on trouve que les trois méthodes K++, K++R et Sample deviennent plus performantes que les deux méthodes Var-Part et SB.

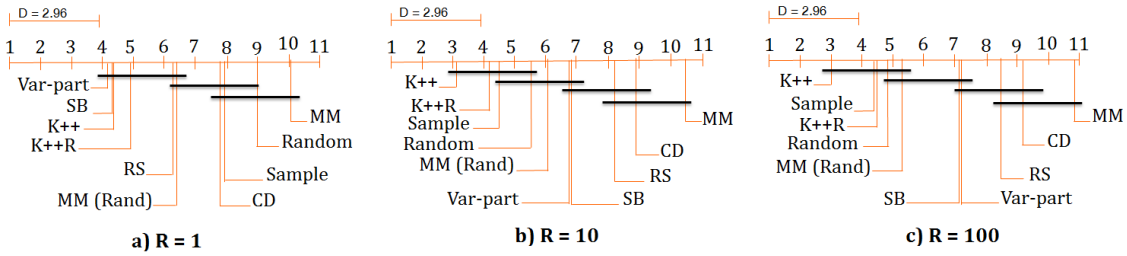


FIGURE 4.11 – Comparaison des méthodes d'initialisation (précédées par Conditional Info) en utilisant le test de Friedman couplé au test post-hoc de Nemenyi (ALC-MSE en test)

Avec le prétraitement non supervisé : A partir de la figure 4.12, on observe que quel que soit le nombre de réplicates R, les méthodes d'initialisation se rapprochent en termes de performance (la constitution d'un seul groupe suivant le test de Nemenyi). Néanmoins, les deux méthodes SB et K++R restent les premières en classement.

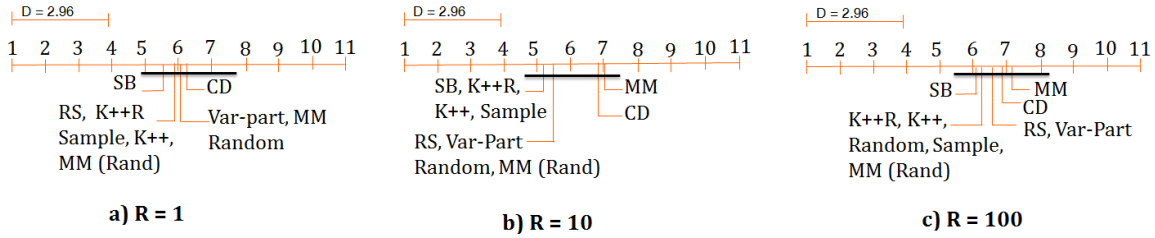


FIGURE 4.12 – Comparaison des méthodes d'initialisation (précédées par RN-BGB) en utilisant le test de Friedman couplé au test post-hoc de Nemenyi (ALC-MSE en test)

4.6.3 Évaluation du compromis

Une bonne méthode d'initialisation suivant le principe des K-moyennes prédictives est celle qui cherche à trouver le bon compromis entre la prédiction et la compacité. Puisqu'il n'existe pas dans la littérature un critère global permettant d'évaluer ce compromis, on s'est basé alors sur le principe du Front de Pareto. Les figures 4.13, 4.14 et 4.15 présentent respectivement, pour les 21 jeux de données, le rang des quatre premières méthodes obtenues pour l'ARI vis-à-vis de la MSE (en utilisant Conditional Info a) et Rank Normalization b)) dans le cas où $R = 1, 10$, et 100. Sur ces figures, plus les points approchent de l'origine des axes, plus la méthode utilisée sur ces jeux de données arrive à réaliser un bon compromis entre les deux critères. Lorsque on utilise Conditional Info comme un prétraitement, nous observons que les deux méthodes qui arrivent à atteindre un bon compromis entre la MSE et l'ARI sont : SB et RS pour $R = 1$, SB et K++R suivie par RS pour $R = 10$ et SB, RS et K++ pour $R = 100$ (voir le graphiques gauche de chaque figure). Lorsqu'on utilise Rank Normalization et Basic Grouping comme un prétraitement nous observons que quel que soit le nombre de réplicates R utilisé, les méthodes d'initialisation ont presque la même performance en termes de MSE. Cependant, les deux méthodes RS et SB ont une meilleure performance en termes d'ARI.

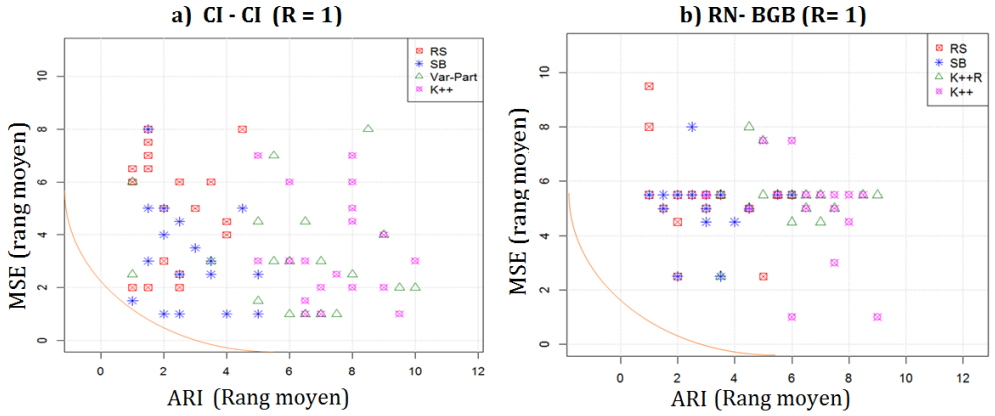


FIGURE 4.13 – Le rang des quatre premières méthodes en termes d'ARI vis-à-vis de la MSE pour $R = 1$ (la ligne orange représente un guide de lecture)

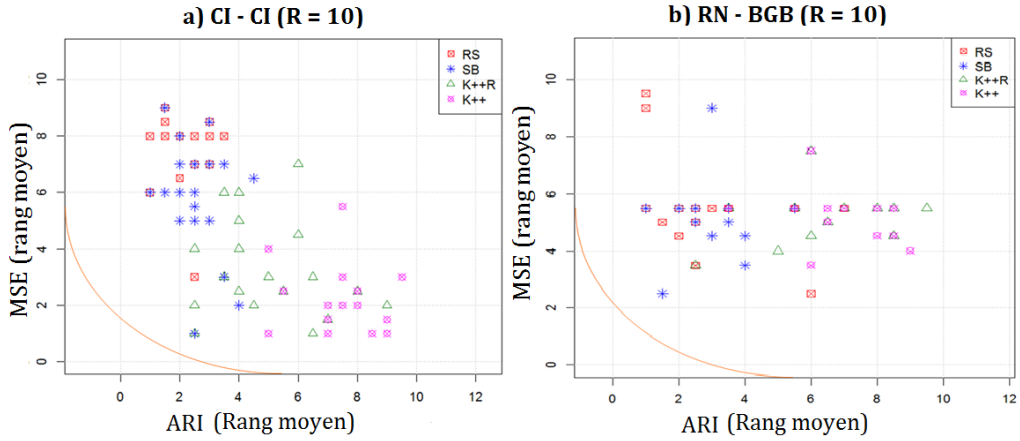


FIGURE 4.14 – Le rang des quatre premières méthodes en termes d'ARI vis-à-vis de la MSE pour $R = 10$ (la ligne orange représente un guide de lecture)

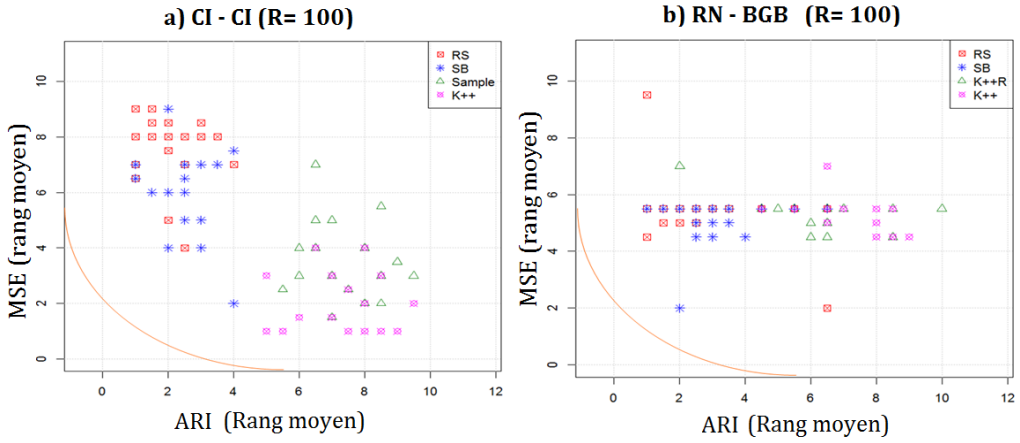


FIGURE 4.15 – Le rang des quatre premières méthodes en termes d'ARI vis-à-vis de la MSE pour $R = 100$ (la ligne orange représente un guide de lecture)

Une autre façon de comparer les méthodes d'initialisation proposées en termes de performance est de mesurer les écarts de performances (en termes de valeurs) entre les méthodes proposées et une méthode prise en référence. Par exemple, pour le critère à maximiser ARI et pour une méthode A, la formule utilisée est la suivante : $(ARI(A) - ARI(ref)/ARI(ref)) \times 100$. Une valeur positive signifie que la méthode A a de meilleure performance en termes d'ARI vis à vis de la méthode ref et vice versa. La méthode de référence choisie dans cette étude comparative est la méthode qui fournit une bonne performance en termes de description à savoir, KMean++. La figure 4.16 présente la comparaison en pourcentage de l'ALC-ARI et de l'ALC-MSE des méthodes d'initialisation proposées vis-à-vis de la méthode de référence KMean++ lorsque l'algorithme est exécuté 1 fois (partie gauche de la figure), 10 fois (partie milieu de la figure) et 100 fois (partie droite de la figure). Dans cette figure, les valeurs de l'ALC-ARI et de l'ALC-MSE représentent une moyenne sur les 21 jeux de données. Nous constatons ici que lorsque l'algorithme des K-moyennes n'est exécuté qu'une seule fois, les trois méthodes d'initialisation proposées ont de meilleures performances en termes d'ARI par rapport à la méthode de référence et une performance en termes de MSE presque similaire à la méthode KMean++ (par exemple, pour la méthode RS, on trouve 19% en ALC-ARI et −3% en ALC-MSE). Lorsque l'algorithme est exécuté plusieurs fois (*i.e.*, $R \in \{10, 100\}$), la méthode de référence KMean++ devient meilleure en termes de MSE (e.g, -11% de l'ALC-MSE pour S-Bisecting lorsque $R = 100$) mais elle reste moins performante que les méthodes proposées en termes d'ARI (e.g, 19% de l'ALC-ARI pour S-Bisecting lorsque $R = 100$).

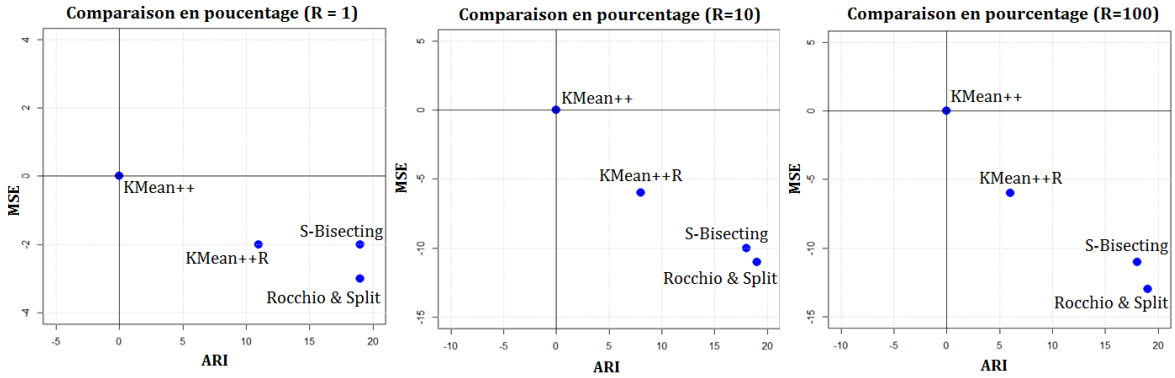


FIGURE 4.16 – Comparaison (en pourcentage) de la méthode KMean++ avec les trois méthodes d'initialisation proposées

Discussion : En se basant sur l'ensemble des résultats obtenus dans cette deuxième partie d'expérimentation, nous constatons que :

- *Pour l'axe de prédiction* : quel que soit le prétraitement utilisé (Conditional Info ou Rank Normalization et/ou basic Grouping), nous remarquons que l'algorithme des K-moyennes standard précédé par la méthode Rocchio-and-Split (RS) fournit de meilleurs résultats en termes de prédiction par rapport à sa performance en utilisant les autres méthodes d'initialisation.
- *Pour l'axe de description* : Lorsque Conditional Info est utilisé, la méthode Var-Part fournit de meilleurs résultats en termes de MSE quand l'algorithme des K-moyennes est exécuté qu'une seule fois ($R = 1$). Quand on augmente la valeur du nombre de réplicates R ($R \in \{10, 100\}$), les trois méthodes K++, K++R et Sample deviennent celles qui fournissent de bons résultats en termes de MSE. Lorsque Rank Normalization et/ou Basic

Grouping [13] est utilisé, les performances en termes de la MSE des méthodes d'initialisation deviennent plus proches les unes des autres.

— *Pour le compromis* : quel que soit le prétraitement utilisé et le nombre de réplicates R, la méthode SB suivie par la méthode RS sont les deux méthodes qui parviennent à établir un certain compromis entre la description et la prédiction. A titre d'exemple, la méthode S-Bisecting perd 10% de la MSE mais gagne en parallèle 20% en ARI vis-à-vis de la méthode KMeans++. Cependant, lorsque le nombre de clusters est élevé la méthode Rocchio-and-split (SB) reste préférable à la méthode S-Bisecting en raison de sa complexité (SB nécessite beaucoup d'effort de calcul).

4.7 Bilan et synthèse

Ce chapitre a présenté l'influence d'une étape d'initialisation supervisée sur la qualité des résultats générés par l'algorithme des K-moyennes standard. Nous avons pu montrer qu'une bonne méthode d'initialisation supervisée a la capacité d'aider cet algorithme à atteindre l'objectif des K-moyennes prédictives (i.e., le compromis entre la prédiction et la compacité). Ce résultat reste inchangé quel que soit le prétraitement utilisé (supervisé ou non supervisé). Dans le cas où le nombre de clusters est égal au nombre de classes (problème de classification supervisée), la méthode K-Mean++R parvient à obtenir de meilleurs résultats en termes de prédiction en une seule exécution (méthode déterministe). Dans le cas où le nombre de clusters est supérieur au nombre de classes, SB et RS sont les meilleures méthodes (parmi les 10 méthodes) arrivant à atteindre un certain compromis entre la prédiction et la compacité. La figure 4.17 présente l'évolution de l'ALC-ARI (moyenne sur les 21 jeux de données) suivant le prétraitement et la méthode d'initialisation utilisé. A partir de cette figure nous remarquons que l'utilisation d'un prétraitement supervisé et d'une méthode d'initialisation supervisée améliore davantage la performance prédictive de l'algorithme des K-moyennes classique (en passant de 0.17 pour RN-BGB et K++ à 0.38 pour CI-CI et RS).

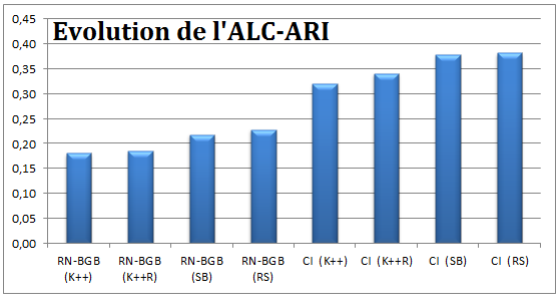


FIGURE 4.17 – Évolution de l'ALC-ARI (en test) suivant le prétraitement et la méthode d'initialisation utilisée

La figure 4.18, quant à elle, présente l'évolution de l'ALC-MSE (moyenne sur les 21 jeux de données) suivant la méthode d'initialisation utilisée en utilisant Conditional Info (partie gauche de la figure) et Rank Normalization\Basic Grouging (partie droite de la figure) comme prétraitement. Ces deux graphiques montrent que les méthodes d'initialisation supervisées ont une

13. Il est à rappeler que le choix du prétraitement utilisé dépend de la nature des variables existant dans le jeu de données : on utilise Rank Normalization pour les variables continues et Basic Grouping pour les variables catégorielles.

performance similaire en termes de MSE à la méthode non supervisée KMean++ lorsque le prétraitement non supervisé est utilisé et une performance moins bonne que celle de la méthode non supervisée.

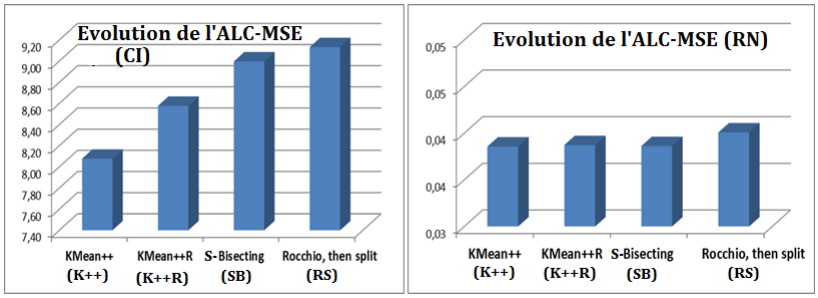


FIGURE 4.18 – Évolution de l'ALC-MSE (en test) suivant la méthode d'initialisation utilisée pour CI (partie à gauche) et pour RN\BGB (partie à droite).

Dans le domaine d'apprentissage automatique, l'évaluation de la qualité des résultats fournis par un algorithme d'apprentissage est une tâche cruciale. Cependant, dans le cadre du clustering prédictif, il n'existe pas de critère global permettant de mesurer la qualité des résultats. Seul le principe du Front de Pareto peut être utilisé pour sélectionner la ou les méthodes qui réalisent le meilleur compromis entre la prédiction et la description (voir Section 4.6.3). De ce fait, la recherche d'un critère d'évaluation global qui permet de mesurer ce compromis s'avère nécessaire. Le chapitre qui suit sera consacré à la recherche de ce critère.

Chapitre 5

Évaluation de la qualité de l'algorithme des K-moyennes prédictives

Sommaire

Ce chapitre a fait l'objet de la publication suivante :

[13] Oumaima Alaoui Ismaili, Vincent Lemaire, Antoine Cornuéjols : « Evaluation of predictive clustering quality », in MBC2, on Model Based clustering and classification (MBC2,2016).

5.1 Introduction

Comme évoqué précédemment dans le chapitre 2 Section 2.5.2, il existe deux types de clustering prédictif. **Le premier type** consiste à discerner un nombre *minimal* de groupes d'instances purs en termes de classes dans le but de prédire ultérieurement la classe des nouvelles instances (voir la figure 5.1). Il s'agit de découvrir *partiellement* la structure interne de la variable cible. Comme pour la classification supervisée, le but majeur de ces algorithmes est de prédire correctement la classe des nouvelles instances. De ce fait, pour évaluer la qualité des résultats issus de ce type d'algorithme, l'un des critères supervisés dédiés à la classification supervisée, tels l'indice de rand Ajusté (ARI) [57] ou la variation d'information (VI) [76], peut être utilisé.

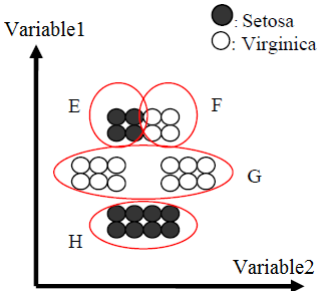


FIGURE 5.1 – Premier type du clustering prédictif

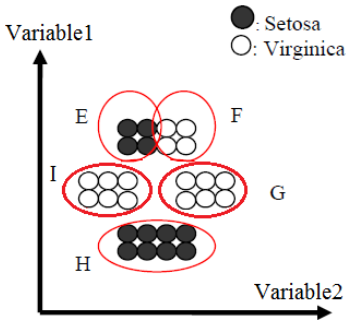


FIGURE 5.2 – Deuxième type du clustering prédictif

Le deuxième type, quant à lui, a pour but de discerner des groupes d'instances compacts, purs en termes de classe et éloignés les uns des autres (voir la figure 5.2). Contrairement à la classification supervisée, les algorithmes appartenant à ce type d'apprentissage cherchent à découvrir la structure interne *complète* de la variable cible. Puis, munie de cette structure, ils cherchent à prédire la classe des nouvelles instances. Dans ce cadre d'étude, aucun axe n'est privilégié par rapport à l'autre (*i.e.*,, la description et la prédiction). Une bonne partition au sens du deuxième type du clustering prédictif est donc celle qui réalise un bon compromis entre la description et la prédiction. Le critère choisi pour mesurer la qualité des résultats issus par les algorithmes du deuxième type du clustering prédictif doit impérativement équilibrer les trois points suivant :

1. Inertie intra-clusters minimale.
2. Inertie inter-clusters maximale.
3. Taux de bonnes classifications maximal.

Ce chapitre traite exclusivement la problématique d'évaluation de la qualité pour l'algorithme des K-moyennes prédictives du deuxième type. Dans la phase d'apprentissage, l'algorithme des K-moyennes prédictives nécessite une évaluation de la qualité à deux niveaux : 1) *Pour le choix de la meilleure partition à K fixé.* En effet, l'algorithme des K-moyennes prédictives converge rarement vers un optimum global. Pour cette raison, pour un nombre fixe de clusters, cet algorithme doit être exécuté plusieurs fois dans le but de choisir, via un critère analytique, la meilleure partition au sens du deuxième type du clustering prédictif, 2) *Pour la sélection du nombre optimal de clusters (K_{opti}).* En effet, l'algorithme des K-moyennes prédictives nécessite une connaissance *a priori* du nombre optimal de clusters ce qui n'est pas une tâche aisée dans la réalité. Pour surmonter ce problème, l'algorithme peut être exécuté plusieurs fois avec différents nombres de

clusters dans le but de sélectionner le nombre optimal de clusters permettant ainsi de mieux décrire la structure interne de la variable cible. Le critère utilisé dans ce cas doit impérativement être capable de comparer deux partitions ayant des nombres de clusters différents ce qui n'est pas une obligation pour le critère utilisé au premier niveau d'évaluation.

Les critères qui peuvent être utilisés pour choisir la meilleure partition (à K fixé) sont notamment les critères supervisés tel que l'indice de Rand Ajusté "ARI" ou les critères non supervisés tel que l'erreur quadratique moyenne "MSE". Le choix du critère à utiliser aura donc un impact direct sur les résultats : la meilleure partition va donc en dépendre. Pour pouvoir effectuer ce choix, il est important tout d'abord de connaître l'influence de l'utilisation d'un des critères (supervisé ou non supervisé) sur les résultats obtenus. Dans ce contexte d'étude, le bon critère à utiliser est celui qui conduit soit à une amélioration significative au niveau des deux axes (*i.e.,* la description et la prédiction) vis-à-vis des résultats obtenus par les autres critères ou soit à une amélioration significative sur l'un des deux axes (par exemple, l'axe de prédiction si le critère supervisé ARI est utilisé) et à une détérioration très légère (voire aucune détérioration) de l'autre axe (voir Figure 5.3). La section 5.2.1 de ce chapitre présente une étude expérimentale permettant de répondre à cette question.

Pour le deuxième niveau d'évaluation, les critères supervisés et non supervisés existant dans la littérature n'arrivent pas dans tous les cas à sélectionner le nombre optimal de clusters permettant de mieux découvrir la structure interne de la variable cible. Par exemple, pour les critères non supervisés, cette incapacité s'illustre essentiellement dans le cas de la présence des régions denses possédant au moins deux classes. En effet, la majorité de ces critères se basent sur une métrique permettant de mesurer la proximité entre les instances indifféremment de leurs classes d'appartenance. Par conséquent, des instances de différentes classes peuvent être vues comme similaires si elles sont proches en termes de distance. Pour tenter de résoudre cette problématique, ce chapitre propose une version supervisée de l'indice de Davies-Bouldin, noté **SDB**. Cet indice est basé sur une nouvelle mesure de similarité 'supervisée' permettant de relier la proximité des instances en termes de distance à leur classe d'appartenance : *deux instances sont considérées comme similaires si et seulement si elles sont proches en termes de distance et appartiennent à la même classe.* Le lecteur pourra trouver une description plus détaillée de cette problématique dans la section 5.3 de ce chapitre.

Le reste de ce chapitre est organisé comme suit : la section 5.2.1 a pour but de tester l'impact de l'utilisation d'un critère supervisé ou non supervisé sur les résultats obtenus lors du choix de la meilleure partition. Cette étude expérimentale donne une indication de la capacité de ces critères à réaliser un bon compromis entre la description et la prédiction (pour un nombre fixe de clusters). La section 5.2.2 présente une discussion sur les cas où les critères supervisés et non supervisés se montrent incapables de sélectionner le nombre optimal de cluster. Ceci peut être vu comme un point de départ vers la recherche d'un nouveau critère d'évaluation pour le clustering prédictif. Dans ce contexte, la section 5.3 présente la nouvelle version supervisée de l'indice de Davies-Bouldin, notée SDB (Supervised Davies-Bouldin). Cet indice est basé sur une nouvelle mesure de similarité supervisée présentée dans la section 5.3.2. Finalement, avant de conclure ce chapitre dans la section 5.5, quelques études expérimentales seront menées dans la section 5.4 afin de prouver la capacité du critère modifié à bien mesurer le compromis entre la description et la prédiction.

5.2 Évaluation de la qualité du deuxième type du clustering prédictif

5.2.1 Influence du choix de la meilleure partition

Dans la phase d'apprentissage, l'algorithme des K-moyennes prédictive converge rarement vers un optimum global. De ce fait, pour un nombre fixe de clusters, cet algorithme doit être exécuté $R > 1$ fois (voir la boucle **Pour** de l'algorithme 7) dans le but de choisir la meilleure partition au sens du clustering prédictif du deuxième type.

Entrée
- Un ensemble de données D, où chaque instance X_i est décrite par un vecteur de d dimensions et par une classe $Y_i \in \{1, \ldots, J\}$.
- Le nombre de clusters souhaité, noté K.

Début
1) Prétraitement des données.
2) Initialisation des centres.
Pour un nombre fixé de partitions, noté **R faire**
Répéter
3) *Affectation :* générer une nouvelle partition en assignant chaque instance X_i au groupe dont le centre est le plus proche.

$$X_i \in C_k \forall j \in 1, \ldots, K \qquad k = min_j \parallel X_i - \mu_j \parallel$$

avec μ_k est le centre de gravité du cluster C_k.
4) *Représentation :* calculer les centres associés à la nouvelle partition

$$\mu_k = \frac{1}{N_k} \sum_{X_i \in C_k} X_i$$

jusqu'à ce que (convergence de l'algorithme)
Fin Pour
5) Choix de la meilleure partition parmi les R partitions.
6) Attribution des classes aux clusters formés.
7) Prédiction de la classe des nouvelles instances.
Fin

Sortie
- Chaque cluster est représenté par un prototype qui possède la même prédiction de classe.
- Chaque cluster est associé à une description donnée par le biais de langage B.
- L'inertie intra-clusters est minimale (l'homogénéité des instances est maximale).
- L'inertie inter-clusters est maximale (la similarité entre les clusters est minimale).
- Le taux de bonnes classifications est maximal.

Algorithme 7 – Algorithme des K-moyennes prédictives du deuxième type

Il est à rappeler qu'une bonne partition au sens du deuxième type du clustering prédictif est celle qui réalise un bon compromis entre la description et la prédiction. Les trois points à respecter lors de l'évaluation de la qualité des résultats sont notamment la compacité, la séparabilité et la pureté des clusters en termes de classe. Dans ce contexte d'étude, les critères existant dans la littérature permettant de choisir la meilleure partition sont les critères supervisés tels que l'ARI ou les critères non supervisés tels que la MSE. Cependant, les critères supervisés privilégient principalement l'axe de prédiction tandis que les critères non supervisés privilégient principalement l'axe de description. Le choix du critère à utiliser aura donc un impact direct sur la qualité des résultats au sens du clustering prédictif : la meilleure partition va en dépendre. Suivant ce raisonnement, il est naturel de se demander quel est le critère (supervisé ou non supervisé) le plus adéquat à utiliser dans notre cadre d'étude ?

Dans ce contexte d'étude, un bon critère (supervisé ou non supervisé) est défini comme celui qui conduit :

— soit à une amélioration significative au niveau des deux axes vis-à-vis des résultats obtenus par les autres critères d'évaluation (*i.e.,* les résultats obtenus via ce critère ont tendance à suivre la flèche 3 de la figure 5.3 si les deux critères utilisés pour évaluer l'axe description et l'axe de prédiction sont à maximiser).

— soit à une amélioration significative au niveau d'un axe (par exemple, au niveau de l'axe de prédiction si le critère supervisé ARI est utilisé) et à une détérioration très légère (voire aucune détérioration) au niveau de l'autre axe : les résultats obtenus pour la meilleure partition via le critère ont tendance à suivre la flèche 1 de la figure 5.3 si un critère supervisé tel que l'ARI est utilisé pour le choix, ou bien de suivre la flèche 2 de la figure 5.3 si un critère non supervisé tel que MSE est utilisé.

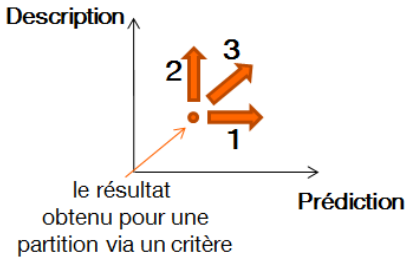


FIGURE 5.3 – L'influence de l'utilisation d'un critère supervisé ou non supervisé sur les résultats.

Pour être en mesure de connaître l'influence du choix du critère d'évaluation sur la qualité des résultats et donc connaître le critère (ARI ou MSE) le plus adéquat à utiliser pour choisir la meilleure partition, une étude expérimentale est alors menée en utilisant plusieurs jeux de données de l'UCI. Cette étude expérimentale nous permet également d'avoir une idée sur le degré de la corrélation entre les deux critères ARI et MSE. Dans cette étude, pour chaque jeu de données, le nombre de clusters est varié entre J et $J+10$, J étant le nombre de classes à prédire. Pour un nombre fixe de clusters, l'algorithme des K-moyennes est exécuté 100 fois dans le but de choisir la meilleure partition en utilisant soit l'ARI soit la MSE. La méthode d'initialisation utilisée à ce stade est la méthode qui garantit un bon compromis entre la prédiction et la description à savoir, S-Bisecting (voir Chapitre 4 Section 4.6).

La partie gauche (respectivement droite) des figures 5.4, 5.5, 5.6, 5.7 et 5.8 présente respectivement les valeurs de l'indice ARI (respectivement MSE) lorsque le choix de la meilleure partition est effectué à l'aide de la MSE (les courbes rouges de la figure) ou à l'aide de l'ARI (voir les courbes bleues de la figure) pour les jeux de données Wine, Hepatitis, Breast, Horsecolic et Segmentation. Le lecteur pourra trouver les résultats sur d'autres jeux de données dans l'annexe D de ce mémoire.

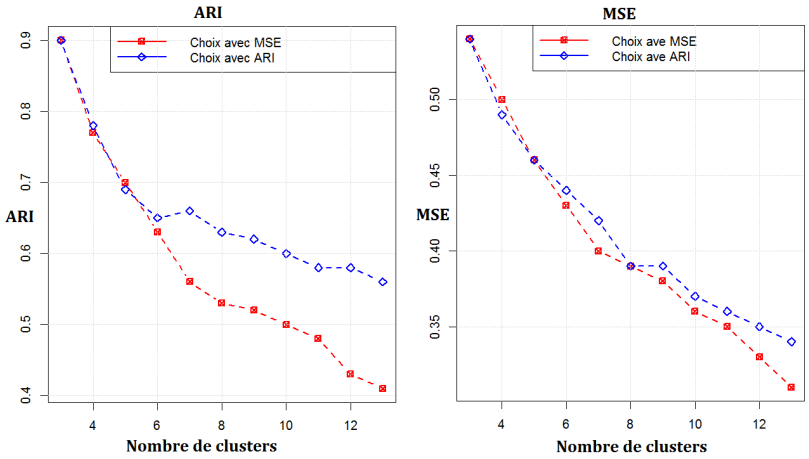


FIGURE 5.4 – L'évolution des courbes de l'ARI (partie gauche) et de la MSE (partie droite) selon le critère utilisé pour choisir la meilleure partition pour le jeu de données **Wine**

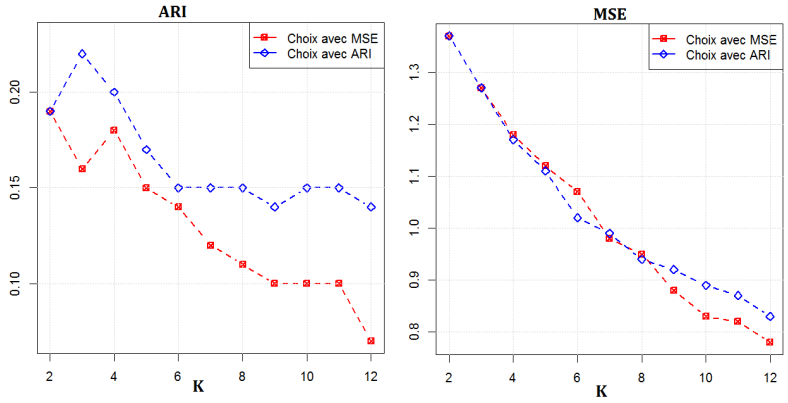


FIGURE 5.5 – L'évolution des courbes de l'ARI (partie gauche) et de la MSE (partie droite) selon le critère utilisé pour choisir la meilleure partition pour le jeu de données **Hepatitis**

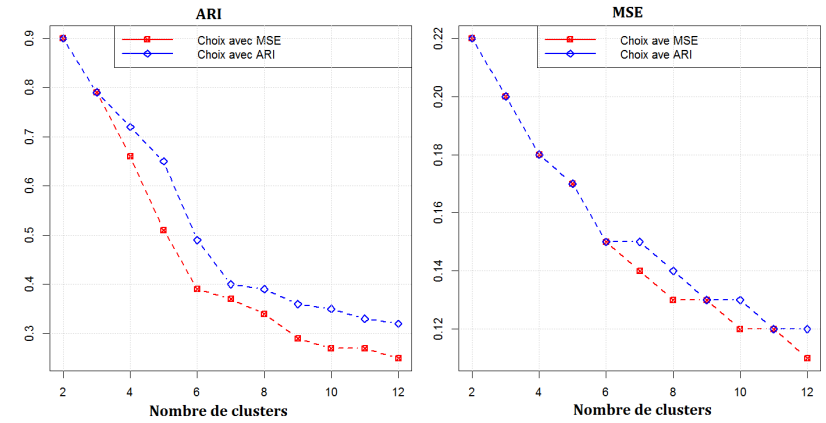


FIGURE 5.6 – L'évolution des courbes de l'ARI (partie gauche) et de la MSE (partie droite) selon le critère utilisé pour choisir la meilleure partition pour le jeu de données **Breast**.

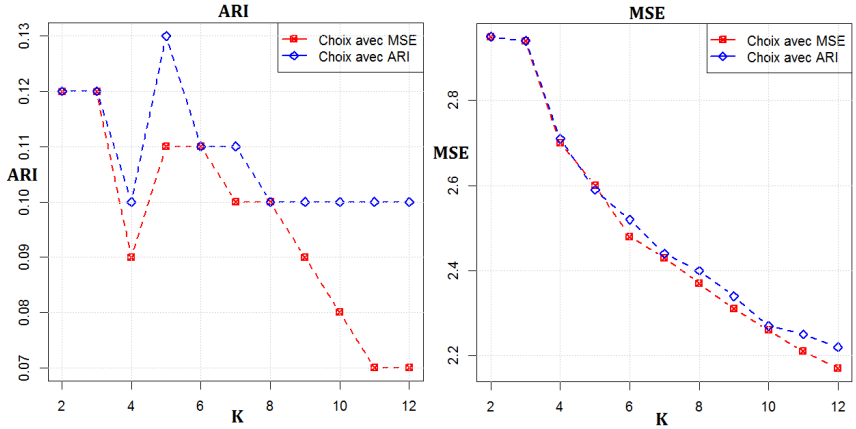


FIGURE 5.7 – L'évolution des courbes de l'ARI (partie gauche) et de la MSE (partie droite) selon le critère utilisé pour choisir la meilleure partition pour le jeu de données **Horsecolic**.

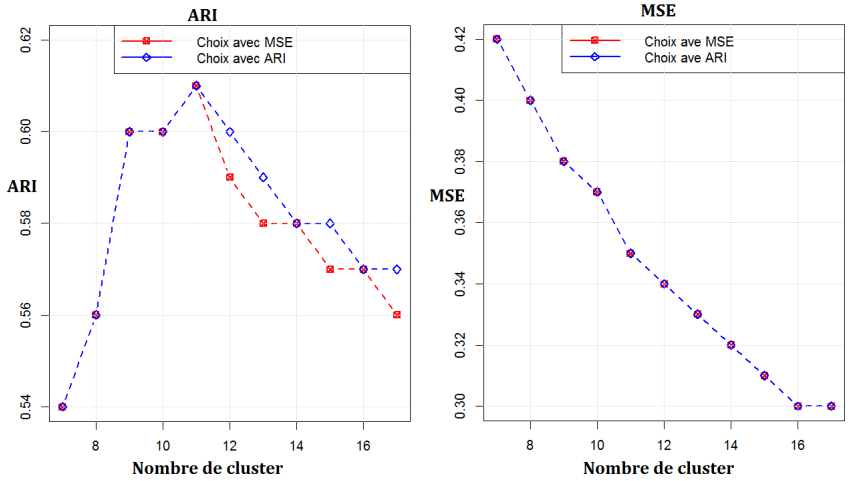


FIGURE 5.8 – L'évolution des courbes de l'ARI (partie gauche) et de la MSE (partie droite) selon le critère utilisé pour choisir la meilleure partition pour le jeu de données **Segmentation**

Les résultats expérimentaux présentés dans ces figures 5.4, 5.5, 5.6, 5.7 et 5.8 prouvent que les deux critères ARI et la MSE ne sont pas forcément corrélés : une amélioration au niveau d'un critère n'implique pas forcement de détérioration au niveau de l'autre critère. À titre d'exemple, la figure 5.5 montre que le choix de la meilleure partition en utilisant l'ARI engendre une amélioration significative au niveau de l'ARI avec une amélioration très légère au niveau de la MSE dans le cas où $k \in \{2, \ldots, 8\}$.

Le critère le plus adéquat parmi l'ARI et la MSE pour choisir la meilleure partition dans le cadre du clustering prédictif du deuxième type semble être (sur ces jeux de données) le critère ARI. En effet, celui-ci permet d'améliorer significativement les performances prédictives du modèle vis-à-vis des résultats obtenus en utilisant le critère non supervisé MSE (à titre d'exemple, pour le jeu de données Wine dans le cas où $K \in \{7, \ldots, 13\}$, voir la courbe bleue de la partie droite de la figure 5.4). De plus, aucune détérioration (par exemple, pour le jeu de données Segmentation, voir la partie droite de la figure 5.8) ou bien une détérioration très légère (par exemple, voir la courbe rouge pour le jeu de données Breast présenté dans la partie droite de la figure 5.6) au niveau de l'axe de description est introduite.

5.2.2 Choix du nombre optimal de clusters

L'algorithme des K-moyennes prédictives nécessite une connaissance a priori du nombre optimal du cluster (voir Algorithme 7). Or, il est très difficile dans la réalité de connaître à l'avance, pour chaque jeu de données, ce nombre optimal. Pour remédier à ce problème, l'algorithme des K-moyennes prédictives doit être exécuté plusieurs fois avec différents nombres de clusters dans le but de sélectionner le nombre de clusters permettant ainsi de découvrir au mieux la structure interne de la variable cible.

Pour le choix du nombre optimal de clusters, le critère recherché doit être capable de sélectionner la partition découvrant la structure interne "complète" de la variable cible. Il s'agit ici de pouvoir comparer deux partitions ayant un nombre de clusters différents au sens du deuxième type du clustering prédictif. Les critères supervisés et non supervisés existant dans la littérature permettant de comparer des partitions avec différents nombre de clusters n'arrivent pas toujours à détecter le nombre de clusters optimal. En effet, les critères proposés dans le cadre de la classification supervisée privilégient principalement l'axe de prédiction par rapport à l'axe de description. D'une part, l'utilisation des critères tels que la précision (ACC) et l'aire sous la courbe de ROC (AUC) pour mesurer la qualité des résultats s'avère inappropriée dans ce cadre d'étude : l'augmentation du nombre de clusters induit souvent une amélioration, ou une stagnation, au niveau de la performance prédictive du modèle. Par conséquent, la partition optimale sélectionnée par ce type de critère peut contenir un nombre de clusters très grand par rapport au "réel" nombre.

D'autre part, l'utilisation des critères d'évaluation tels que l'indice de rand ajusté (ARI) ou les critères basés sur la théorie d'information comme la variation d'information (VI), s'avère également insuffisante : lorsque deux sous-groupes différents d'instances de même classe sont proches comme illustré dans la figure 5.9 (les deux sous-groupes de la classe rouge situés au milieu de la figure), ces deux critères cherchent à les fusionner ensemble. Dans le cas extrême, ces deux sous-groupes peuvent être également fusionnés même s'ils sont très éloignés l'un de l'autre.

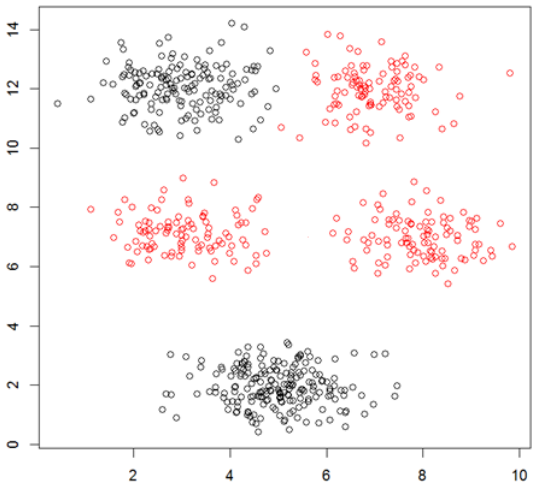


FIGURE 5.9 – Exemple illustratif d'incapacité des critères supervisés à mesurer le compromis description/prédiction

Enfin, les critères non supervisés proposés dans la littérature pour évaluer la qualité des résultats issus des algorithmes du clustering traditionnel privilégient principalement l'axe de description. L'utilisation de tels critères dans le cadre du clustering prédictif s'avère inappropriée. En effet, l'incapacité des critères non supervisés à mesurer la qualité des résultats issus des algorithmes du clustering prédictif s'illustre principalement dans le cas de la non corrélation entre les clusters et les classes. C'est le cas de la présence de plus de deux classes dans au moins une des régions denses. À titre d'exemple, le jeu de données présenté dans la figure 5.10 possède deux régions denses caractérisées par la présence des deux classes (rouge et noire). Pour cet exemple, il est clair que la partition optimale suivant ces critères est celle qui contient le nombre de régions denses comme nombre de clusters (*i.e.,* 4 clusters). Par conséquent, deux groupes formés dans ce cas vont contenir des instances de classes différentes ce qui conduit à une détérioration au niveau de la prédiction.

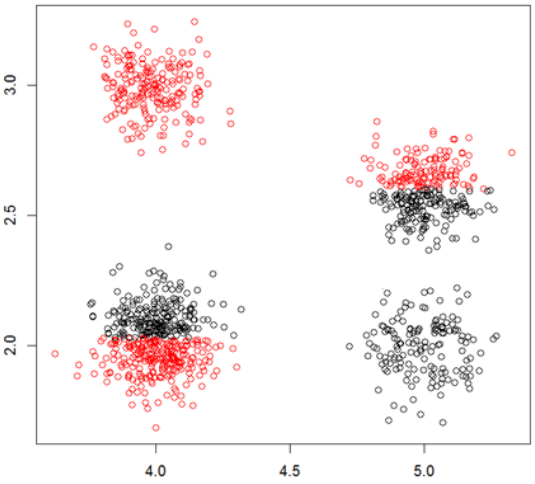


FIGURE 5.10 – Exemple illustratif d'incapacité des critères non supervisés à mesurer le compromis description/prédiction

Ceci est dû au fait que la majorité de ces critères se basent sur une mesure de similarité (ou une distance) qui évalue la proximité entre les instances indifféremment de leurs classes d'appartenance. De ce fait, deux instances proches en termes de distance vont être considérées comme similaires bien qu'elles appartiennent à des classes différentes. On en conclut que, la recherche d'un critère permettant de mesurer le compromis entre la description et la prédiction s'avère nécessaire.

5.2.3 Vers la recherche d'un critère d'évaluation

Pour mesurer le compromis entre la description et la prédiction, le front de Pareto [20] peut être utilisé. C'est une technique qui permet de résoudre les problèmes d'optimisation multi-critères. Elle consiste à chercher un ensemble de solutions dites non dominées, parmi lesquelles on ne peut pas décider si l'une est meilleure que l'autre, aucune ne permet systématiquement de trouver l'optimum pour tous les critères. Cet ensemble est nommé l'ensemble de Pareto.

Soient $f_1, \ldots, f_Z$ des critères à minimiser et x_1 et x_2 deux solutions potentielles au problème multi-critères. La solution x_1 est dite dominée x_2 si :

$$\forall i \quad f_i(x_1) \leq f_i(x_2)$$
$$\text{avec au moins un } i \text{ tel que } f_i(x_1) < f_i(x_2)$$

- La solution x_1 est dite faiblement non dominée, s'il n'existe pas de solution x_2 telle que : $\forall i, f_i(x_1) \leq f_i(x_2)$.

- La solution x_1 est dite fortement non dominée, s'il n'existe pas de solution x_2 telle que : $f_i(x_1) \leq f_i(x_2)$ avec au moins un i tel que $f_i(x_1) < f_i(x_2)$

La figure 5.11 illustre le concept de dominance dans le cas d'un problème d'optimisation bicritère, un problème où on cherche à minimiser deux fonctions f_1 et f_2. Dans ce cas, les solutions représentées par les points A et B dominent la solution représentée par le point C. Par contre, les solutions représentées par les points A , B et D ne sont dominées par aucune solution. Les points représentatifs de ces solutions non dominées constituent le front de Pareto.

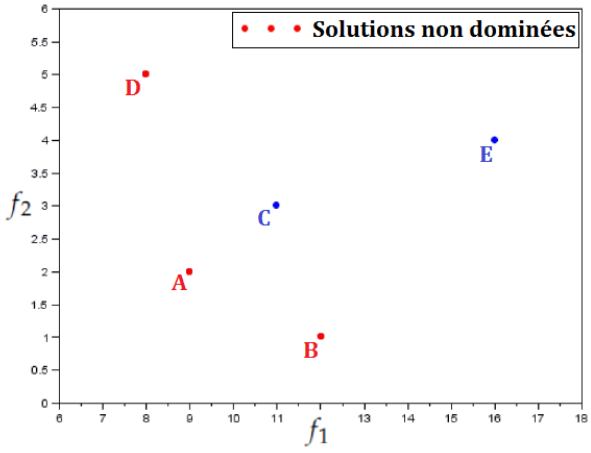


FIGURE 5.11 – Illustration de la notion de dominance pour un cas bicritère, f_1 et f_2 sont deux fonctions à minimiser. La solution A domine les solutions C et E

Dans notre cadre d'étude, les deux critères à optimiser sont notamment un critère supervisé (*e.g.,* ARI, VI) qui mesure la performance prédictive du modèle en question et un critère non

supervisé (*e.g.,* Indice Davies-Bouldin [38]) qui mesure la capacité de celui-ci à discerner des groupes d'instances compacts et éloignés les uns des autres.

À notre connaissance, il n'existe pas dans la littérature de critère analytique qui permette de mesurer la qualité des résultats issus du deuxième type du clustering prédictif. Cependant, certains chercheurs ont intégré dans leurs approches de clustering prédictif des fonctions objectifs cherchant à combiner deux quantités différentes (supervisée et non supervisée). Ces fonctions peuvent donc être un point de départ vers la recherche d'un nouveau critère d'évaluation.

Par exemple, Eick *et al.* [46] ont proposé une fonction objectif à minimiser pour l'algorithme des K-modes supervisé. Cette fonction se compose d'un critère supervisé ($Impurity(X)$) évaluant la capacité du modèle à bien classer les instances et d'un critère non supervisé ($penalty(K)$) pénalisant l'obtention d'un nombre maximal de clusters. La formule mathématique de cette fonction est donnée par l'équation 5.1 :

$$q(X) = Impurity(X) + \beta * penalty(K) \tag{5.1}$$

avec

$$Impurity(X) = \frac{Nombre\ d'instances\ mal\ classes}{N}$$

et

$$penalty(K) == \begin{cases} \sqrt{\frac{K-J}{n}}, & K \geq J \\ 0 & K < J \end{cases}$$

β est un paramètre utilisateur compris entre 0 et 2. Une grande valeur de β implique une large pénalité pour un nombre élevé de clusters. Si on se place dans le cadre du deuxième type du clustering prédictif, l'utilisation de cette fonction pour mesurer le compromis prédiction/description reste insuffisant. En effet, la proximité entre les paires d'instances en termes de distance n'est pas introduite dans celle-ci. De plus, les résultats obtenus via ce critère vont dépendre du choix de la valeur de *beta* qui n'est pas une tâche facile.

Peralta et al. ont proposée dans [85] une fonction objectif pour l'algorithme des K-moyennes supervisé écrite sous forme d'une combinaison convexe entre deux quantités différentes : l'une représente la fonction objectif usuelle de l'algorithme des K-moyennes standard et l'autre représente sa version supervisée. La formule mathématique de cette fonction est donnée par l'équation 5.2

$$J = \sum_{n=1}^{N} \left[\alpha \sum_{k=1}^{K} \sum_{l=1}^{J} \delta_{nk}^{l} \parallel x_n - u_k^l \parallel^2 \rho_k^l + (1-\alpha) \sum_{k=1}^{K} \delta_{nk} \parallel x_n - u_k \parallel^2 \right] \tag{5.2}$$

avec δ_{nk}^l est une fonction indicatrice supervisée qui assigne l'instance x_n au centre u_k^l désignant le centre de gravité du cluster k ayant comme classe l. ρ_k^l est un facteur défini pour les instances de classe l appartenant au cluster k. δ_{nk} est fonction indicatrice non supervisée qui assigne l'instance x_n au cluster k. Finalement, α est un paramètre utilisateur compris entre 0 et 1 gérant l'équilibre entre les deux scores (supervisé et non supervisé) du clustering. Comme pour la fonction proposée par Eick *et al.*, les résultats obtenus via cette fonction vont dépendre du choix de la valeur de α qui n'est pas une tâche facile.

5.3 Proposition d'un indice pour le clustering prédictif (Type 2)

5.3.1 Motivation

Pour mesurer le compromis entre la description et la prédiction, trois points doivent être respectés, à savoir : 1) la minimisation de l'inertie intra-clusters, 2) la maximisation de l'inertie inter-clusters et 3) la maximisation du taux de bonnes classifications. Un bon critère au sens du clustering prédictif est donc celui qui équilibre ces trois points et qui vérifie les contraintes, à savoir :

1. *L'interprétabilité* : le critère doit être facile à interpréter
2. *Le nombre de clusters* : le critère ne doit pas être trop biaisé par le nombre de clusters. En effet, ce critère doit être capable de comparer deux partitions ayant un nombre de clusters différent.
3. *Le nombre d'instances* : le critère ne doit pas être trop biaisé par le nombre d'instances dans chaque cluster. En effet, ce critère doit être capable de mesurer le degré de la compacité, la séparabilité et la pureté dans des clusters de différents effectifs. Par exemple, en se basant sur des proportions.
4. *La complexité* : le critère ne doit pas avoir une complexité trop supérieure à celle de l'algorithme des K-moyennes prédictives utilisé.
5. *Le bruit* : le critère doit être relativement stable en cas de perturbations aléatoires.

Lors de la recherche d'un nouveau critère pour le deuxième type du clustering prédictif, deux voies intuitives peuvent être exploitées, à savoir : *i*) la modification d'un critère supervisé à travers l'intégration d'une mesure qui évalue la proximité des paires d'instances, et *ii*) la modification d'un critère non supervisé à travers l'intégration d'une mesure qui relie la proximité des instances en termes de distance à leurs classes d'appartenance. Dans cette thèse, nous nous intéressons à l'étude de la deuxième voie. Il s'agit ici de modifier le critère d'évaluation Davies-Bouldin (DB) communément utilisé dans le cadre du clustering standard.

La raison de l'incapacité de l'indice de Davies-Bouldin à sélectionner le nombre optimal des clusters dans le cas de la non corrélation entre les classes et les clusters revient au fait que celui-ci utilise une métrique (ou une distance) non supervisée pour mesurer la ressemblance entre les instances. Cette métrique évalue la ressemblance entre instances en se basant sur leur proximité en termes de distance et sans accorder aucune importance à leurs classes d'appartenance. Par conséquent, des instances de classes différentes peuvent être considérées comme similaires si elles sont proches en termes de distance (cas d'une région dense possédant au moins deux classes). Pour permettre à l'indice Davies-Bouldin de surmonter ce problème, une intégration d'une mesure de similarité supervisée s'avère nécessaire. La section suivante propose donc une nouvelle mesure de similarité permettant de prendre en considération l'appartenance des instances aux différentes classes lors de l'évaluation de leur proximité.

5.3.2 Proposition d'une nouvelle mesure de similarité supervisée

Une similarité ou dissimilarité est définie comme étant toute application à valeurs numérique qui permet de mesurer le lien entre les individus d'un même ensemble. Pour une similarité, le lien entre deux individus sera d'autant plus fort que sa valeur est grande. Pour une dissimilarité le lien sera d'autant plus fort que sa valeur de la dissimilarité sera petite.

Définition :

Un opérateur de ressemblance $s : \mathcal{D}\times\mathcal{D} \rightarrow [0,1]$ défini sur l'ensemble d'individus $\mathcal{D} = \{X_i\}_{i=1}^{N}$ est un indice de similarité (ou similarité), s'il vérifie les propriétés suivantes :

1. *Symétrie* : $\forall X_i, X_j \in \mathcal{D} \quad s(X_i, X_j) = s(X_j, X_i)$
2. *Positivité* : $\forall X_i, X_j \in \mathcal{D} \quad s(X_i, X_j) \geq 0$
3. *Maximalité* : $\forall X_i, X_j \in \mathcal{D} \quad s(X_i, X_i) = s(X_j, X_j) \geq s(X_i, X_j)$

Il convient de noter ici que le passage de l'indice de similarité s à la notion duale d'indice de dissimilarité (que nous noterons Sim), est trivial. Étant donné s_{max} la similarité d'une instance avec elle-même ($s_{max} = 1$), il suffit de poser :

$$\forall X_i, X_j \in \mathcal{D} \quad Sim(X_i, X_j) = s_{max} - s(X_i, X_j) \tag{5.3}$$

Proposition d'une nouvelle mesure de dissimilarité

Dans le contexte supervisé, chaque instance $X_i = \{X_{i1}, \ldots, X_{id}\}_{i=1}^{N}$ possède d variables descriptives et une variable qualitative décrivant sa classe d'appartenance $Y_i = f(X_i)$. Pour évaluer la ressemblance entre deux paires d'instances étiquetées, quatre scénarios possibles peuvent être définis :

— **Scénario 1** : Proches (en termes de distance) et de même classe (Figure 5.12 A).
— **Scénario 2** : Proches (en termes de distance) et de classes différentes (Figure 5.12 C).
— **Scénario 3** : Éloignées (en termes de distance) et de même classe (Figure 5.12 B).
— **Scénario 4** : Éloignées (en termes de distance) et de classes différentes (Figure 5.12 D).

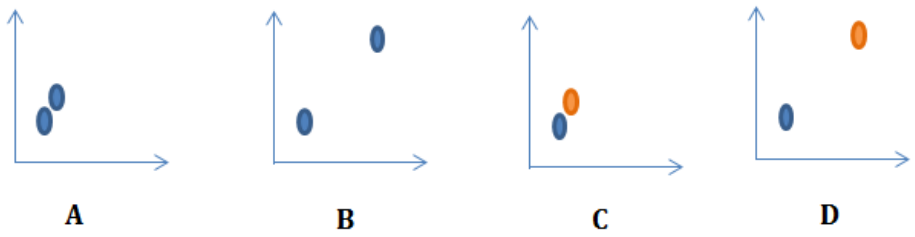


FIGURE 5.12 – Les 4 scénarios illustrant la ressemblance entre deux instances étiquetées

Suivant ces quatre scénarios, il est clair que la forte similarité « $s(X_i, X_j) = 1$ » ou la faible dissimilarité « $Sim(X_i, X_j) = 0$ » va correspondre au premier scénario (Figure 5.12 A) et la faible similarité (ou la forte dissimilarité) va correspondre au quatrième scénario (Figure 5.12 D). La mesure de similarité/dissimilarité proposée dans ce contexte doit prendre en compte les quatre scénarios cités ci-dessus.

Soit X_i et X_j deux instances de dimension d dans $\mathcal{D}$ appartenant respectivement à la classe $f(X_i)$ et $f(X_j)$. La nouvelle mesure de dissimilarité $Sim(X_i, X_j)$ qui relie la proximité de X_i et X_j (en termes de distance) à leurs classes d'appartenance est définie comme suit :

$$\forall X_i, X_j \in \mathcal{D} \quad Sim(X_i, X_j) = 1 - \frac{exp(-\delta(X_i, X_j))}{1 + dist(X_i, X_j)^2} \tag{5.4}$$

avec δ est une fonction indicatrice présentée comme suit :

$$\delta(X_i, X_j) = \begin{cases} 0 & \text{si} \quad f(X_i) = f(X_j) \\ 1 & \text{si} \quad f(X_i) \neq f(X_j) \end{cases} \tag{5.5}$$

Il est à noter que la vraie classe $f(X_i)$ de l'instance X_i peut être remplacée par la classe prédite $\hat{f}(X_i)$ selon le besoin.

$dist(X_i, X_j)$ est la distance Euclidienne donnée par la formule suivante :

$$dist(X_i, X_j) = \|X_i - X_j\|_2 = \sqrt{\sum_{l=1}^{d}(X_{il} - X_{jl})^2}, \quad \forall X_i, X_j \in \mathcal{D} \tag{5.6}$$

Il est à noter que dans le cas de grandes dimensions, la quantité $dist(X_i, X_j)^2$ sera très grande. Ceci peut induire une stagnation au niveau de la mesure de similarité $s(X_i, X_j) = \frac{exp(-\delta(X_i, X_j))}{1+dist(X_i, X_j)^2}$. Pour pallier ce problème, l'utilisation des données normalisées sera utile pour une diminution de la quantité $dist(X_i, X_j)^2$. De plus, la mesure suivante peut être utilisée.

$$dist(X_i, X_j)^2 = \sum_{l=1}^{d} \frac{(X_{il} - X_{jl})^2}{d} \tag{5.7}$$

La quantité $s(X_i, X_j) = \frac{exp(-\delta(X_i, X_j))}{1+dist(X_i, X_j)^2}$ est bien une mesure de similarité qui vérifie les trois propriétés citées ci-dessus, à savoir la symétrie, la positivité et la maximalité. Elle prend ses valeurs dans l'intervalle $[0, 1]$ tout comme la mesure de dissimilarité $Sim(X_i, X_j)$ présentée dans l'équation 5.4 :

— $\forall X_i, X_j \in \mathcal{D} \quad Sim(X_i, X_j) = 0 \Leftrightarrow dist(X_i, X_j) = 0 \quad \textbf{ET} \quad X_i$ et X_j ont la même classe.
— $\forall X_i, X_j \in \mathcal{D} \quad Sim(X_i, X_j) = 1 \Leftrightarrow dist(X_i, X_j) \to \infty$.

À l'aide de cette nouvelle mesure de dissimilarité, deux instances sont considérées comme similaires, si et seulement si, elles sont proches en termes de distance et appartiennent à la même classe. Cependant, lorsqu'elles appartiennent à des classes différentes, leur proximité en termes de distance $\frac{1}{dist(A, X_i)^2+1}$ est pénalisée par le terme $exp(-1)$. À titre d'exemple, la partie milieu de la figure 5.13 présente l'influence de la classe sur les résultats obtenus par la mesure proposée. La courbe noire (*respectivement*, la courbe bleue) présente les valeurs de la mesure de dissimilarité proposée entre une instance A et d'autres instances de l'espace (voir la partie gauche de la figure 5.13) lorsque toutes les instances appartiennent à la même classe (respectivement, les instances ont une classe différente de celle de l'instance A).

Cette figure montre clairement l'impact de la classe sur les résultats obtenus par la mesure proposée. La partie droite de la figure 5.13, quant à elle, présente la distance euclidienne entre l'instance A et les autres instances de l'espace présenté dans la partie gauche de la figure 5.13. Visuellement, on remarque que la courbe obtenue par la distance euclidienne a la même allure que les courbes obtenues par la mesure proposée.

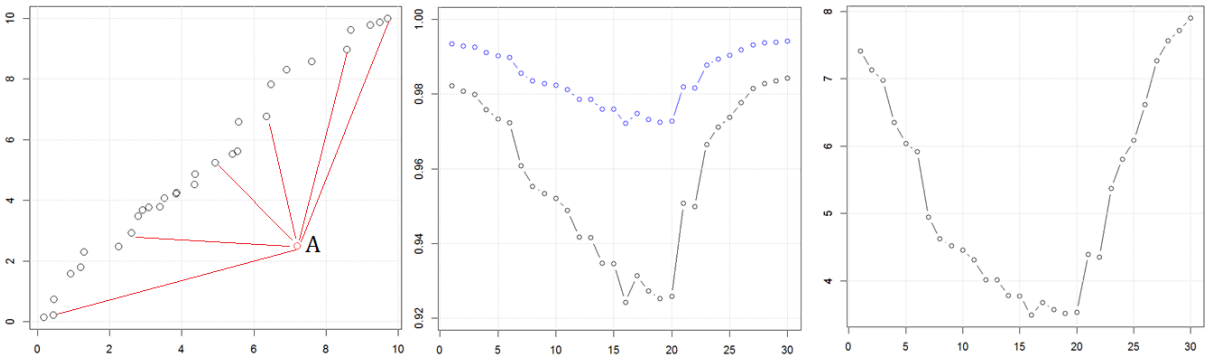


FIGURE 5.13 – Différence entre la mesure proposée (figure du milieu) et la distance Euclidienne (figure à droite).

5.3.3 La version supervisée de l'indice de Davies-Bouldin (SDB)

Avant d'intégrer la nouvelle mesure de dissimilarité proposée ci-dessus dans l'indice de Davies-Bouldin , il est important de le définir dans son contexte.

Rappel

L'indice Davies-Bouldin (DB) [38] traite chaque cluster individuellement et cherche à mesurer à quel point il est similaire au cluster qui lui est le plus proche. L'indice DB est décrit par la formule suivante :

$$DB = \frac{1}{K}\sum_{k=1}^{K} max_{1\leq k\neq t\leq K}\left\{\frac{S_k + S_t}{M_{kt}}\right\} \tag{5.8}$$

Pour chaque cluster $k \in \{1, \ldots, K\}$ de la partition, l'indice DB cherche le cluster t $(t \neq k)$ qui maximise la quantité R_{kt}, décrite par la formule suivante :

$$R_{kt} = \frac{S_k + S_t}{M_{kt}} \tag{5.9}$$

S_k mesure le degré de la compacité du cluster k. Elle représente la moyenne des distances entre les observations du cluster k et leur centre de gravité G_k. La formule mathématique de S_k est donnée par l'équation 5.10.

$$S_k = \left(\frac{1}{N_k}\sum_{i=1}^{N_k} \parallel X_i - G_k \parallel_p\right)^{\frac{1}{p}} \tag{5.10}$$

La quantité M_{kt}, quant à elle, mesure le degré de la séparabilité entre les deux clusters k et t. Elle représente donc la distance entre le centre de gravité des deux clusters (voir équation 5.11).

$$M_{kt} = \parallel G_k - G_t \parallel_p \tag{5.11}$$

La mesure R_{kt} vérifie les trois propriétés suivantes :

1. $R_{kt} \geq 0$
2. **Si** $S_k \geq S_t$ **et** $M_{kt} = M_{tm}$ **alors** $R_{kt} > R_{tm}$

3. **Si** $S_k = S_m$ **et** $M_{kt} \leq M_{tm}$ **alors** $R_{kt} > R_{tm}$

À partir de ces propriétés, on constate que plus l'indice DB est minimal, plus les clusters formés sont compacts et éloignés les uns des autres.

Dans le cadre du clustering prédictif, une bonne partition au sens du clustering prédictif est celle qui fournit des groupes d'instances compacts, purs en termes de classe et éloignés les uns des autres. La nouvelle version du critère Davies-Bouldin, nommée **SDB** (Supervised Davies-Bouldin) doit être capable d'équilibrer les trois points suivants :

— L'inertie intra-clusters minimale (compacité)
— L'inertie inter-clusters maximale (séparabilité)
— Le taux de bonnes classifications est maximal (prédiction)

L'algorithme des K-moyennes prédictives cherche à former dans la phase d'apprentissage des groupes d'instances compacts, purs en termes de classes et éloignés les uns des autres dans le but de prédire ultérieurement la classe des nouvelles instances. Dans notre cadre d'étude, la compacité et la pureté en termes de classes peuvent être évaluées simultanément en intégrant la nouvelle mesure de similarité donnée par l'équation dans la quantité S_k comme suit :

$$S_k = \frac{1}{N_k} \sum_{i=1}^{N_k} Sim(X_i, G_k) \tag{5.12}$$

avec

$$Sim(X_i, G_k) = 1 - \frac{exp(-\delta_1(X_i, G_k))}{1 + dist(X_i, G_k)^2} \tag{5.13}$$

Le score S_k mesure le degré de ressemblance des instances du cluster k avec leur centre de gravité G_k comme le montrent l'équation 5.12 et la figure 5.14. Dans le cadre supervisé, cette ressemblance est évaluée en respectant les 4 scénarios discutés dans la section 5.3.2.

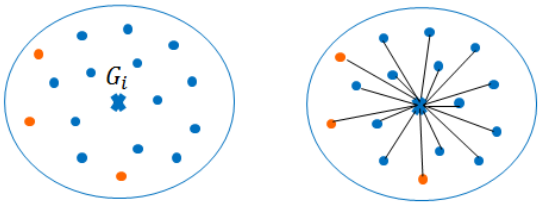


FIGURE 5.14 – Evaluation de la compacité et de la pureté pour un cluster donné.

L'utilisation de la nouvelle mesure de dissimilarité nécessite une connaissance des étiquettes des instances en question. Pour évaluer la pureté en termes de classes dans le cluster k, la vraie classe $f(X_i)$ de chaque instance X_i est comparée à la classe prédite (ou induite) pour le cluster k. Cette classe prédite est associé au centre de gravité G_k du cluster k, notée $\hat{f}(G_k)$. La fonction indicatrice δ_1 qui compare les étiquettes dans la nouvelle mesure de dissimilarité est donnée par l'équation 5.14.

$$\delta_1(X_i, G_k) = \begin{cases} 0 & \text{si} \quad f(X_i) = \hat{f}(G_k) \\ 1 & \text{si} \quad f(X_i) \neq \hat{f}(G_k) \end{cases} \tag{5.14}$$

La mesure de compacité S_k prend ses valeurs dans l'intervalle $[0, 1]$. Cependant, $S_k = 0$ si le cluster k est formé d'une seule instance et $S_k = 1$ si les instances qui le forment sont très éloignées les unes des autres. De ce fait, on constate que plus S_k est petite plus le cluster k est compact et pur en termes de classe.

Pour la séparabilité des clusters, la nouvelle mesure de dissimilarité est utilisée dans le but d'évaluer la ressemblance entre les centres de gravité comme le montre la figure 5.15. Cette ressemblance est mesurée en utilisant la nouvelle mesure de dissimilarité donnée par l'équation 5.15. Il est à rappeler que les étiquettes associées aux centres de gravité sont éventuellement les classes prédites pour les clusters.

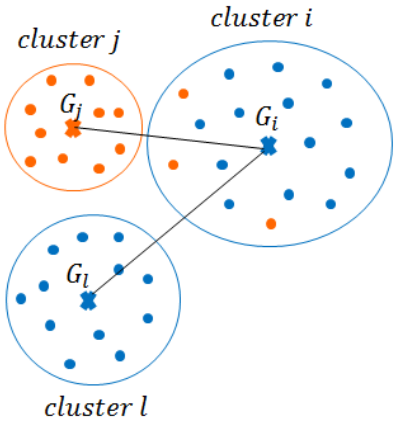


FIGURE 5.15 – Evaluation d'éloignement d'un cluster par rapport aux autres.

$$M_{kt} = Sim(G_k, G_t) = 1 - \frac{exp(-\delta_2(G_k, G_t))}{1 + dist(G_k, G_t)^2} \tag{5.15}$$

avec

$$\delta_2(G_k, G_t) = \begin{cases} 0 & \text{si} \quad \hat{f}(G_k) = \hat{f}(G_t) \\ 1 & \text{si} \quad \hat{f}(G_k) \neq \hat{f}(G_t) \end{cases} \tag{5.16}$$

La mesure de séparabilité M_{kt} prend ses valeurs dans l'intervalle $[0, 1]$. Elle est égale à zéro si $dist(G_k, G_t) = 0$ et les deux clusters ont la même classe prédite ($\hat{f}(G_t) = \hat{f}(G_k)$). Elle est égale à 1 si et seulement si $dist(G_k, G_t) \to \infty$. De ce fait, plus M_{kt} est grande plus les deux clusters sont éloignés les uns des autres.

La version supervisée de l'indice Davies-Bouldin, nommée SDB prend ses valeurs dans l'intervalle $[0, +\infty[$ et est définie comme suit :

$$SDB = \frac{1}{K} \sum_{k=1}^{K} max_{1 \leq k \neq t \leq K} \left\{ \frac{S_k + S_t}{M_{kt}} \right\} \tag{5.17}$$

avec les scores S_k et M_{kt} sont donnés respectivement par les équations 5.14 et 5.15. Comme l'indice de Davies-Bouldin standard, SDB est un critère à minimiser. Plus SDB est proche de 0 plus les groupes appris sont compacts, purs en termes de classes et éloignés les uns des autres.

Récapitulatif

Davies- Bouldin (DB)	**Davies-Bouldin supervisé (SDB)**
L'indice DB s'écrit sous la forme suivant : $DB = \frac{1}{K}\sum_{k=1}^{K} max_{1\leq k\neq t\leq K}\{R_{kt}\}$	L'indice SDB s'écrit sous la forme suivant : $SDB = \frac{1}{K}\sum_{k=1}^{K} max_{1\leq k\neq t\leq K}\{R_{kt}\}$
avec	avec
$R_{kt} = \frac{S_k+S_t}{M_{kt}}$	$R_{kt} = \frac{S_k+S_t}{M_{kt}}$
La compacité	**La compacité**
$S_k = \left(\frac{1}{N_k}\sum_{i=1}^{N_k} \parallel X_i - G_k \parallel_p\right)^{\frac{1}{p}}$	$S_k = \frac{1}{N_k}\sum_{i=1}^{N_k} Sim(X_i, G_k)$
	avec
X_i est une instance de dimension d	$Sim(X_i, G_k) = 1 - \frac{exp(-\delta_1(X_i,G_k))}{1+dist(X_i,G_k)^2}$
G_k est le centre de gravité du cluster k	$\delta_1(X_i, G_k) = \begin{cases} 0 & \text{si} \quad f(X_i) = \hat{f}(G_k) \\ 1 & \text{si} \quad f(X_i) \neq \hat{f}(G_k) \end{cases}$
La séparabilité	**La séparabilité**
$M_{kt} = \parallel G_k - G_t \parallel_p$	$M_{kt} = Sim(G_k, G_t) = 1 - \frac{exp(-\delta_2(G_k,G_t))}{1+dist(G_k,G_t)^2}$
	avec
	$\delta_2(G_k, G_t) = \begin{cases} 0 & \text{si} \quad \hat{f}(G_k) = \hat{f}(G_t) \\ 1 & \text{si} \quad \hat{f}(G_k) \neq \hat{f}(G_t) \end{cases}$
	$\hat{f}(G_k)$ est la classe prédite pour le cluster k
La plage de variation	**La plage de variation**
$[0; +\infty[$	$[0; +\infty[$

5.4 Expérimentation

Afin de vérifier la capacité de la version supervisée de l'indice Davies-Bouldin à mesurer le compromis entre la description et la prédiction et donc mesurer la qualité des résultats issus par les algorithmes de clustering prédictif, nous allons utiliser différents jeux de données : *i*) des jeux

de données contrôlés de petite dimension. Le nombre de variables descriptives dans ce cas est fixé à 2. Pour ces jeux de données, la structure interne de la variable cible est connue à l'avance. L'objectif ici est donc de connaître la capacité du critère modifié (SDB) à bien sélectionner le nombre optimal de clusters dans le cas de la corrélation et la non corrélation entre les clusters et les classes, *ii*) des jeux de données simulés de grandes dimensions. La structure interne de la variable cible pour chacun de ces jeux de données est également connue a priori. L'objectif ici est de mesurer la capacité du critère SDB à fournir de bons résultats (*i.e.,* compromis entre la description et la prédiction) y compris le cas de la grande dimensionnalité, *iii*) un jeu de données de grande dimension de l'UCI dont la structure interne de la variable cible n'est pas connue. Dans ce cas, on cherche à tirer des conclusions sur la capacité de SDB vis-à-vis du critère non supervisé DB et du critère supervisé ARI.

5.4.1 Sur des jeux de données contrôlés

Cas 1 : Non corrélation des classes et des clusters

Comme évoqué précédemment, dans le cas de la non corrélation entre les classes et les clusters, les critères non supervisés tel que DB n'arrivent pas à détecter le nombre de clusters optimal au sens du clustering prédictif. À titre d'exemple, le jeu de données présenté dans la partie gauche de la figure 5.16 est caractérisé par la présence de deux régions denses possédant deux classes (rouge et noire). Visuellement, pour ce jeu de données, la partition optimale au sens du clustering prédictif est celle qui contient 6 groupes. La partie milieu de la figure présente les valeurs des deux critères DB et SDB en fonction du nombre de clusters. Il est à signaler que les partitions sont obtenues en utilisant l'algorithme des K-moyennes standard précédé par la méthode d'initialisation S-Bisecting. Le critère supervisé SDB arrive à détecter la partition optimale pour ce jeu de données jouet tandis que le critère non supervisé DB n'arrive pas à détecter le nombre exacte des régions denses (*i.e.,* 4 régions).

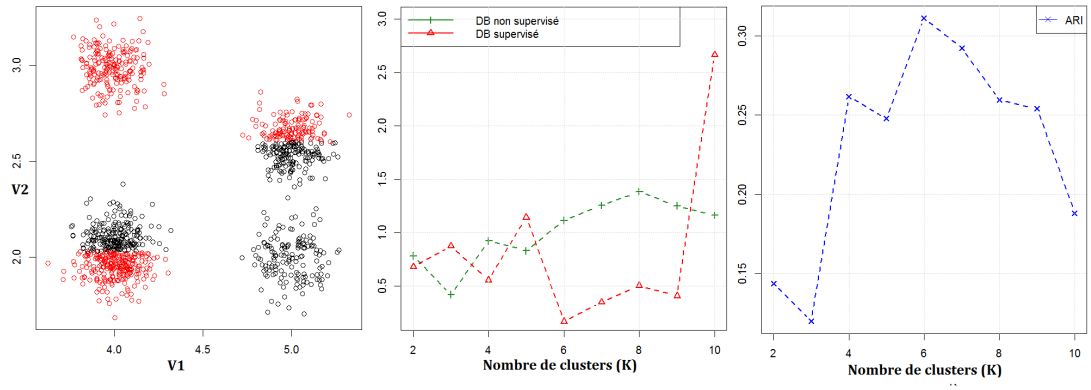


FIGURE 5.16 – Premier jeu de données jouet

L'une des propriétés importantes que le critère d'évaluation proposé doit posséder est la résistance au bruit qui peut exister dans les données. Le jeu de données présenté dans la partie gauche de la figure 5.17 est caractérisé par la présence de deux classes dont chacune possède deux sous-groupes différents. Un bon critère doit pouvoir les détecter malgré le bruit d'étiquetage existant. La partie milieu (respectivement la partie gauche) de la figure 5.17 présente les valeurs du critère SDB (respectivement, DB) lorsqu'on ajoute dans chaque classe 5%, 10%,20%,30 et 40% de bruits. Les résultats obtenus montrent que le critère modifié SDB arrive facilement à

détecter, pour ce jeu de données, le nombre optimal de clusters quel que soit la quantité du bruit intégrée. Cependant le critère DB n'arrive pas à atteindre l'objectif recherché (*i.e.,* la détection du nombre optimal de clusters qui est dans ce cas 4 clusters).

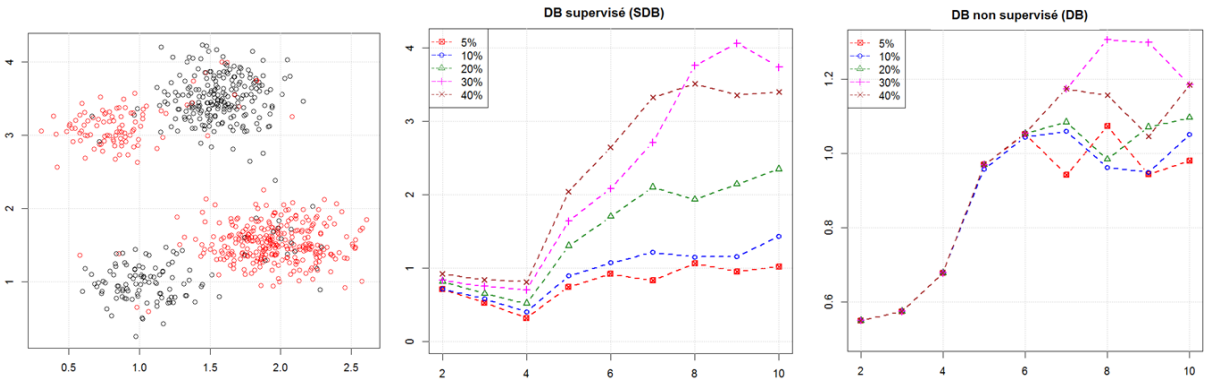


FIGURE 5.17 – Cas d'existence du bruit dans les données.

Cas 2 : Corrélation entre les classes et les clusters

Dans le cas de la corrélation entre les classes et les clusters, les critères non supervisés deviennent alors plus adaptés pour détecter la meilleure partition. Cependant, dans le cas où des sous-groupes différents de même classe sont proches, les critères supervisés tel que l'ARI n'arrivent pas à détecter le nombre optimale de clusters au sens du clustering prédictif puisqu'ils cherchent plutôt à optimiser l'axe de prédiction tout en ignorant l'axe de description. À titre d'exemple, pour les deux jeux de données situés dans la partie gauche des deux figures 5.18 et 5.19, le critère ARI n'arrive pas à détecter le nombre optimal de clusters au sens du clustering prédictif (voir respectivement la partie droite des deux figures 5.18 et 5.19). Le critère non supervisé DB arrive à détecter le nombre optimal du premier jeu de données (voir la courbe verte du graphique situé au milieu des deux figures). Le critère modifié SDB, quant-à-lui, arrive à sélectionner le nombre de clusters optimal dans les deux cas (voir la courbe rouge du graphique au milieu des deux figures).

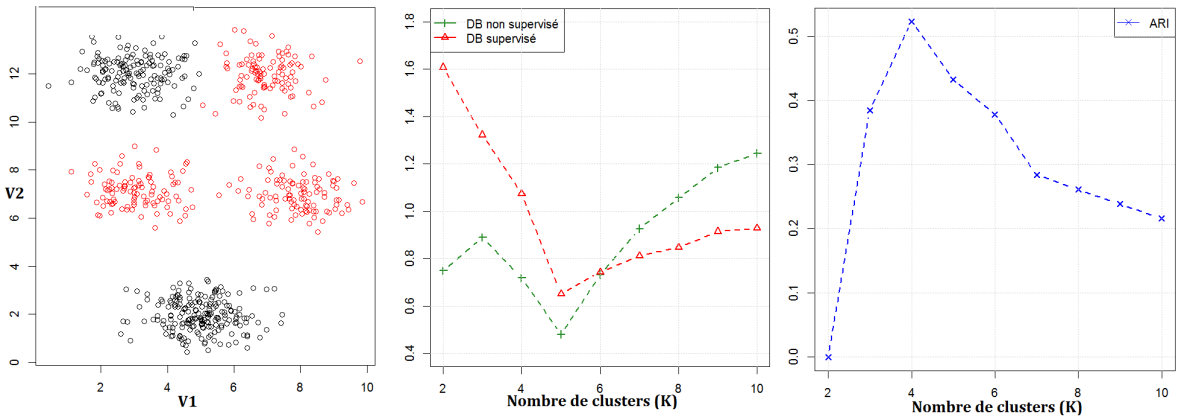


FIGURE 5.18 – Deuxième jeu de données jouet

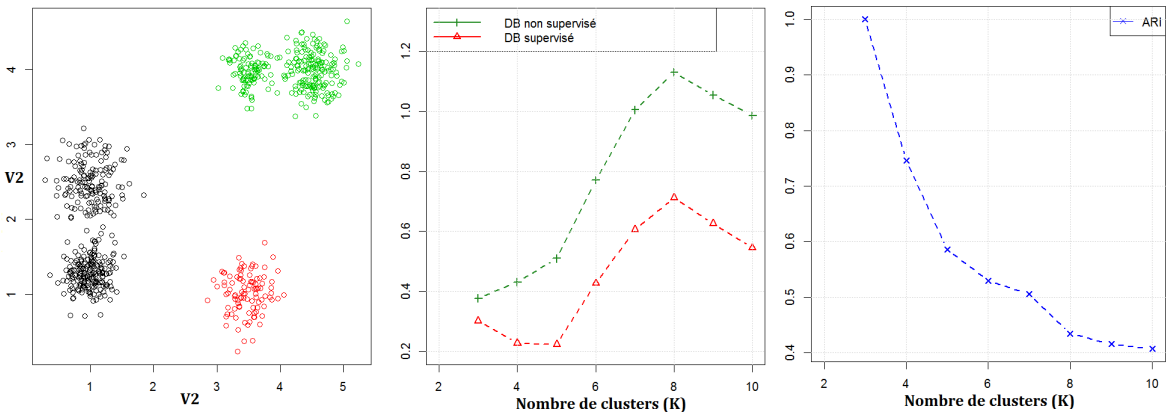


FIGURE 5.19 – Troisième jeu de données jouet

5.4.2 Sur des bases de données simulées de grandes dimensions

Pour valider le comportement du critère modifié vis-à-vis de l'évaluation du compromis description/prédiction dans le cas de la grande dimensionnalité, nous allons simuler quelques jeux de données de façon à obtenir une connaissance *a priori* sur la structure interne de la variable cible. La méthodologie suivie pour obtenir les jeux de données présentés dans ce qui suit est la suivante :

1. générer K_{opti} centres provisoires dans l'espace.
2. Pour chaque centre k_i, générer N_k instances de façon à ce qu'elles soient très proches de leur centre de gravité provisoire.
3. Répéter l'étape 2. pour le reste des centres provisoires.
4. Pour chaque groupe, attribuer une classe.

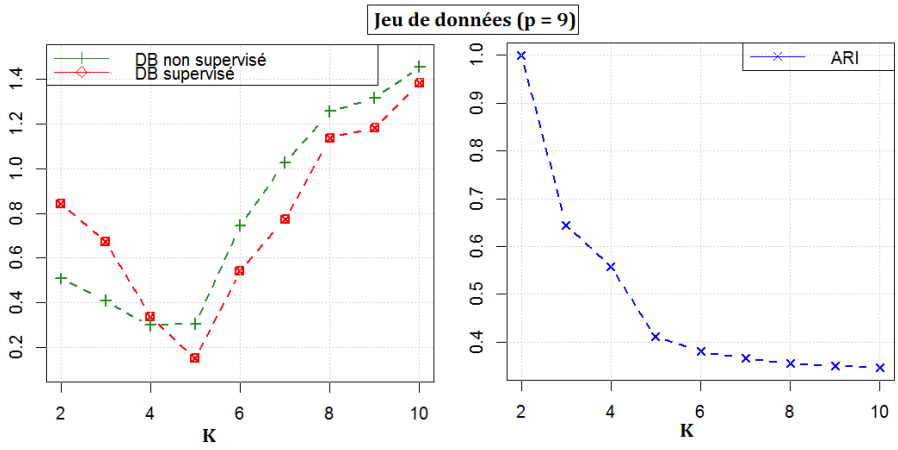


FIGURE 5.20 – Jeu de données contenant 9 variables descriptives, 3 classes et 5 sous-groupes à découvrir

Le premier jeu de données simulé est un jeu de données caractérisé par la présence de 825 instances, 9 variables descriptives et une variable à prédire contenant trois classes à prédire. La première classe contient 2 sous-groupes différents tandis que la deuxième classe contient 3 sous-groupes (*i.e.*, $Kopti = 5$). La figure 5.20 présente les valeurs des critères en fonction du nombre

de clusters. On constate que le critère supervisé ARI n'arrivent pas à détecter le nombre optimal de clusters tandis que les deux critères DB et SDB arrivent à le détecter.

Le deuxième jeu de données simulé est un jeu de données caractérisé par la présence de 765 instances, 9 variables descriptives et une variable possédant deux classes à prédire dont la première contient 3 sous-groupes et la deuxième contient deux sous-groupes (*i.e.,* $Kopti = 5$). Les résultats présentés dans la figure 5.21 montrent que les deux critères ARI et DB n'arrivent pas à détecter le nombre optimal de clusters, tandis que le critère modifié SDB arrive facilement à le détecter.

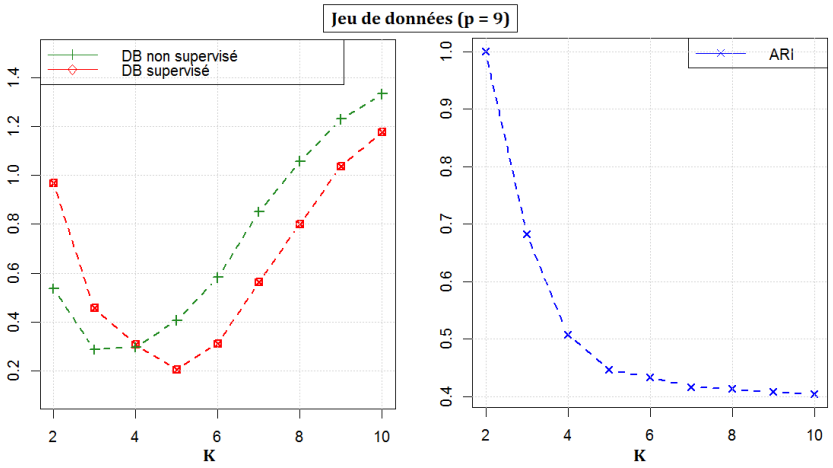


FIGURE 5.21 – Jeu de données contenant 9 variables descriptives, 2 classes et 5 sous-groupes à découvrir

Pour le troisième jeu de données, nous avons augmenté la dimensionnalité. Ce dernier est caractérisé par la présence de 2376 instances, 20 variables descriptives et une variable à prédire contenant 2 classes dont chacune possède deux sous-groupes (*i.e.,* $Kopti = 4$). La figure 5.22 montre que le critère ARI est incapable de détecter le nombre optimal de clusters tandis que les critères DB et SDB y arrivent.

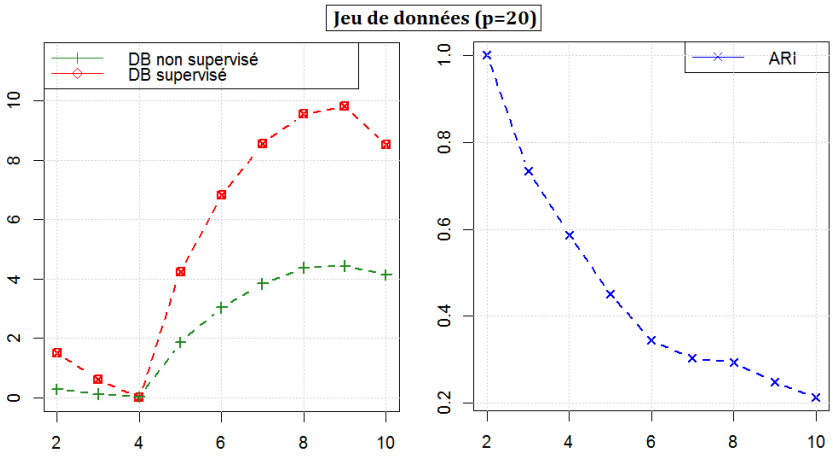


FIGURE 5.22 – Jeu de données contenant 20 variables descriptives, 2 classes et 4 sous-groupes à découvrir

5.4.3 Sur des données de l'UCI

D'après les résultats expérimentaux obtenus dans les sections précédentes, on remarque que le nombre optimal de clusters détecté par la version supervisée de l'indice de Davies-Bouldin, ne doit pas être inférieur au nombre optimal détecté par le critère supervisé ARI **et** par l'indice standard de Davies-Bouldin. En effet, le critère ARI peut fusionner deux sous-groupes différents s'ils sont de la même classe et le critère DB peut fusionner deux sous-groupes proches de classes différentes.

Afin de montrer davantage la capacité du critère SDB à bien détecter le nombre optimal de clusters (au sens du clustering prédictif) et donc détecter la partition qui réalise le bon compromis entre la description et la prédiction, nous allons mener une étude sur la base de données Adult de l'UCI. Cette base est constituée de 48842 instances, 15 variables descriptives et une variable à prédire contenant 2 classes ("more" et "less").

La figure 5.23 présente les valeurs des critères SDB (partie gauche), DB (partie milieu) et ARI (partie droite) en fonction du nombre de clusters. Dans cette étude expérimentale, le nombre de clusters varie de $J = 2$ (J : nombre de classes) jusqu'à 10. Les partitions sont obtenues en utilisant toujours l'algorithme des K-moyennes précédé par le prétraitement Rank normalization pour les variables continues et Basic Grouping pour les variables catégorielles et par la méthode d'initialisation S-Bisecting. On constate que SDB et ARI détectent le même nombre de clusters tandis que l'indice DB indique qu'il n'existe pas une structure interne à découvrir dans la variable cible. Ceci peut être expliqué par le cas de la non corrélation entre les clusters et les classes comme déjà évoqué.

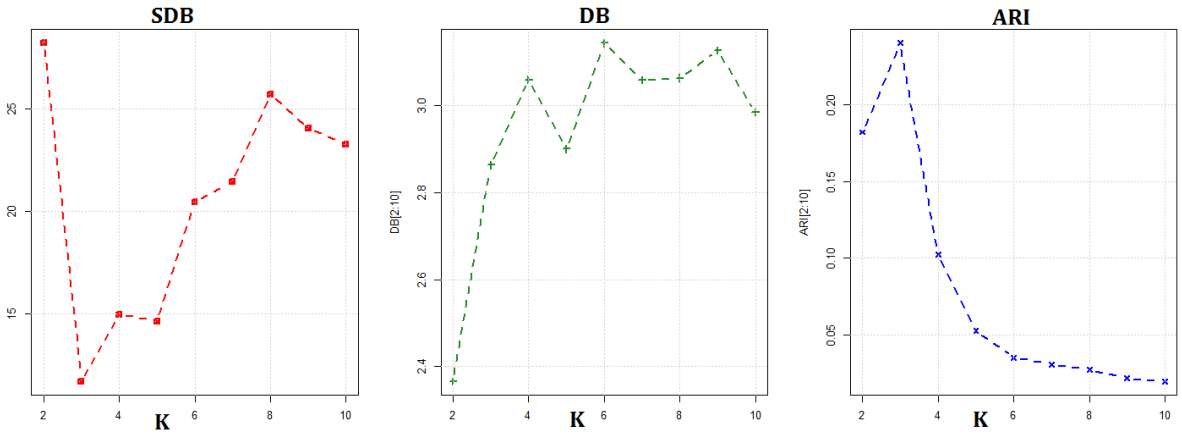


FIGURE 5.23 – Les valeurs des trois critères d'évaluation pour le jeu de données Adult en utilisant Rank Normalization et Basic grouping

5.5 Bilan

Ce chapitre a présenté une version supervisée de l'indice Davies-Bouldin, nommée SDB (Supervised Davies-Bouldin). Cet indice est basé sur une nouvelle mesure de similarité supervisée permettant d'établir une certaine relation entre la proximité des instances en termes de distance et leurs classes d'appartenance. Deux instances sont considérées comme similaires suivant cette nouvelle mesure, si et seulement si, elles sont proches en termes de distance **et** appartiennent à la même classe. Grâce à cette nouvelle mesure, la version supervisée de l'indice de Davies-Bouldin

arrive à surmonter le problème de la non corrélation entre les clusters et les classes. Les résultats expérimentaux ont montré que l'indice SDB arrive à bien détecter le nombre optimal de clusters permettant de mieux découvrir la structure interne de la variable cible par rapport au critère non supervisé DB (voir le tableau 5.1).

Cas	Données	DB	SDB
Jeux de données contrôlés	Jeu 1 (Figure 5.16)	×	✓
	Jeu 2 (Figure 5.17)	×	✓
	Jeu 3 (Figure 5.18)	✓	✓
	Jeu 4 (Figure 5.19)	×	✓
Jeux de données de grands dimensions	Jeu 1 (Figure 5.20)	✓	✓
	Jeu 2 (Figure 5.21)	×	✓
	Jeu 3 (Figure 5.22)	✓	✓
UCI	Adult (Figure 5.23)	×	✓

TABLE 5.1 – Comparaison (échec/succès) des deux critères DB et SDB

Afin de connaître d'avantage la capacité du critère modifié "indice de Davies-Bouldin Supervisé" à mesurer le compromis ente la description et la prédiction et donc mesurer la qualité des résultats issus par l'algorithme des K-moyennes prédictives, le chapitre suivant fera l'objet d'une étude expérimentale sur plusieurs jeux de données. Cette étude expérimentale est divisée en deux grandes parties. La première partie est consacrée à la comparaison des performances de l'algorithme des K-moyennes prédictives du premier type avec les performances des algorithmes de clustering prédictif les plus répandus dans la littérature. L'algorithme des K-moyennes prédictives utilisé dans ce cas englobe les différentes étapes supervisées discutées dans les deux derniers chapitres 2 et 3 (*i.e.,* l'étape du prétraitement des données et l'étape d'initialisation des centres). La deuxième partie, quant-à-elle, est consacrée à l'algorithme des K-moyennes prédictives du deuxième type. Cette partie expérimentale cherche à prouver la capacité de cet algorithme (en utilisant l'indice SDB pour sélectionner le nombre optimal de clusters) à bien découvrir la structure interne globale de la variable cible.

Chapitre 6

Synthèse et Conclusion

6.1 Introduction

L'objectif de cette thèse est la recherche d'un modèle d'apprentissage "interprétable" capable de décrire et de prédire d'une manière simultanée. Ce genre de modèle est connu sous le nom du clustering prédictif. Pour atteindre cet objectif, nous avons choisi de modifier l'algorithme des K-moyennes afin de le rendre un bon prédicteur tout en préservant sa faculté à bien décrire les données. Les chapitres 3 et 4 ce mémoire ont montré respectivement que la supervision de l'étape de prétraitement des données et de l'étape d'initialisation des centres a aidé cet algorithme à atteindre l'objectif du clustering prédictif.

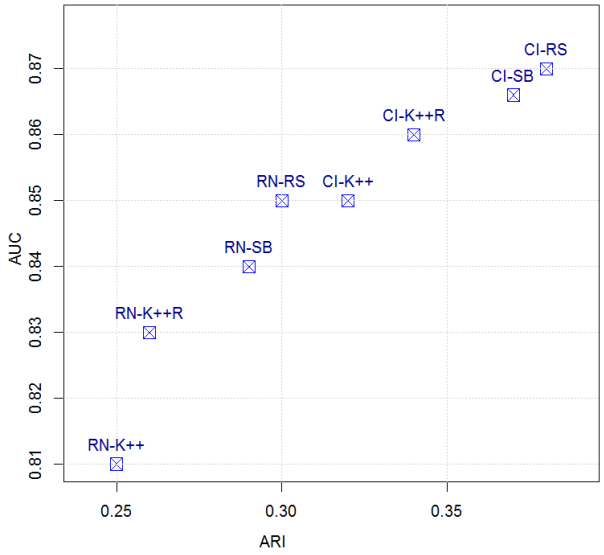


FIGURE 6.1 – Synthèse sur les performances prédictives de l'algorithme des K-moyennes précédé par différentes méthodes de prétraitement et d'initialisation des centres

La figure 6.1 présente la moyenne (obtenue sur 21 bases de données, voir le tableau 4.3 du chapitre 4) des performances prédictives en termes de l'ARI (axe des abscisses) et en termes de l'AUC (axe des ordonnées) de l'algorithme des K-moyennes précédé à chaque fois par différentes méthodes de prétraitements et d'initialisation : **RN** pour Rank Normalization et/ou Basic Grouping, **CI** pour Conditional Info, **K++** pour KMeans++, **K++R** pour KMeans++R, **SB** pour S-Bisecting et **RS** pour Rocchio-And-Split. Pour plus de détails sur ces méthodes, voir les

deux chapitres 3 et 4. L'algorithme des K-moyennes standard est représenté dans la figure 6.1 par RN-K++. Lorsque l'axe de prédiction est privilégié, les résultats présentés dans cette figure montrent que l'algorithme des K-moyennes précédé par le prétraitement supervisé Conditional Info (CI) et la méthode supervisée d'initialisation des centres Rocchio-And-Split (RS) parvient à atteindre de meilleures performances prédictives par rapport à l'algorithme des K-moyennes standard (7.4% en AUC et 32% en ARI).

En ce qui concerne l'axe de description, la figure 6.2 présente les performances de l'algorithme des K-moyennes précédé par Conditional Info et par différentes méthodes d'initialisation des centres en termes de Davies-Bouldin "DB" (voir l'axe des ordonnées) et en termes de Variation d'Information [14] "VI" [76] (voir l'axes des abscisses). Dans ce contexte du compromis, plus la valeur est proche de l'origine des deux axes, plus le modèle parvient à atteindre de bon compromis entre VI et DB. Ces résultats représentent une moyenne sur 21 jeux de données. La figure 6.2 montre que l'algorithme des K-moyennes précédé par la méthode d'initialisation KMeans++ (K++) suivie par la méthode KMeans++R parviennent à atteindre de meilleures performances en termes de DB par rapport aux méthodes d'initialisation supervisées SB et RS.

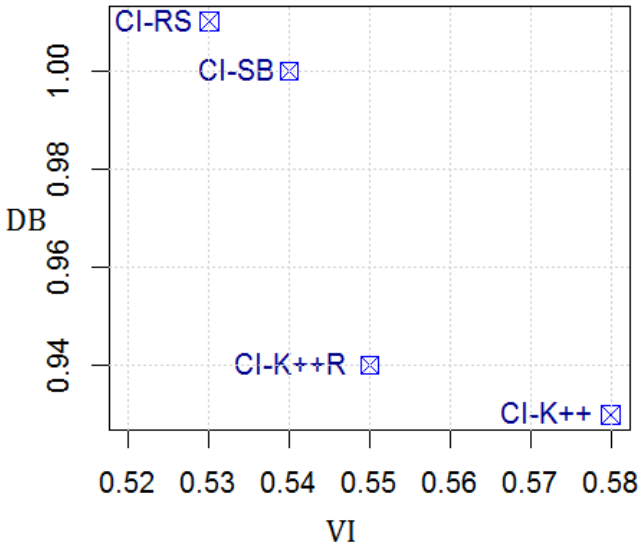


FIGURE 6.2 – Synthèse sur les performances (VI *versus* DB) de l'algorithme des K-moyennes standard précédé par Conditional Info et différentes méthodes d'initialisation

Pour le compromis description-prédiction, les résultats présentées dans la figure 6.2 montrent que l'algorithme des K-moyennes précédé par Conditional Info (prétraitement) et KMeans++R (initialisation) parvient à atteindre un bon compromis entre la description et la prédiction par rapport aux résultats obtenus en utilisant les autres méthodes d'initialisation.

L'objectif de ce chapitre est d'étudier la capacité de l'algorithme des K-moyennes prédictives proposé dans cette thèse à atteindre l'objectif du clustering prédictif. Contrairement aux chapitres précédents où chaque étape est traitée indépendamment des autres, ce chapitre regroupe les différentes méthodes supervisées proposées dans cette thèse (prétraitement, initialisation et critère d'évaluation pour le choix de la meilleure partition) dans l'algorithme des K-moyennes

14. La raison de l'utilisation du critère Variation Information dans ce chapitre est que le critère utilisé dans les chapitres précédents "ARI" est utilisé dans la deuxième partie de la section 6.2 pour choisir la meilleure partition. Pour éviter tout biais, un autre critère de comparaison a donc été choisi. Pour plus de détail sur le critère VI, voir la section 3.3.2 du chapitre 3

prédictives afin de comparer sa performance avec d'autres algorithmes de la littérature. Ce chapitre est divisé en deux grandes parties. La première partie est consacrée au premier type du clustering prédictif (voir Section 6.2). Pour ce type d'algorithmes, l'axe de prédiction est privilégié. Dans ce cadre d'étude, afin d'atteindre notre objectif, nous allons comparer les performances prédictives de l'algorithme des K-moyennes prédictives avec celles obtenues par les algorithmes les plus répandus dans la littérature. La deuxième partie de ce chapitre est consacrée au deuxième type du clustering prédictif (voir Section 6.3). Pour ce type d'algorithmes, aucun axe n'est privilégié par rapport à l'autre. Il s'agit ici de réaliser un bon compromis entre la description et la prédiction sous la contrainte d'interprétation des résultats. Dans cette partie expérimentale, on cherche à connaître, pour un jeu de données illustratif, la capacité de notre algorithme des K-moyennes prédictives à découvrir la structure interne de la variable cible et donc à découvrir les différentes raisons qui peuvent mener à une même prédiction.

Note : L'ensemble des approches présentées dans les sections précédentes ont été codées sur le logiciel R. Des spécifications de codes ont également été fournies à un prestataire afin de faire intégrer les approches proposées dans le logiciel interne **Khiops Ennéade**. Ce dernier est disponible sur le site suivant : www.khiops.predicsis.com. Il est à signaler donc que l'ensemble des résultats obtenus dans cette thèse sont reproductibles.

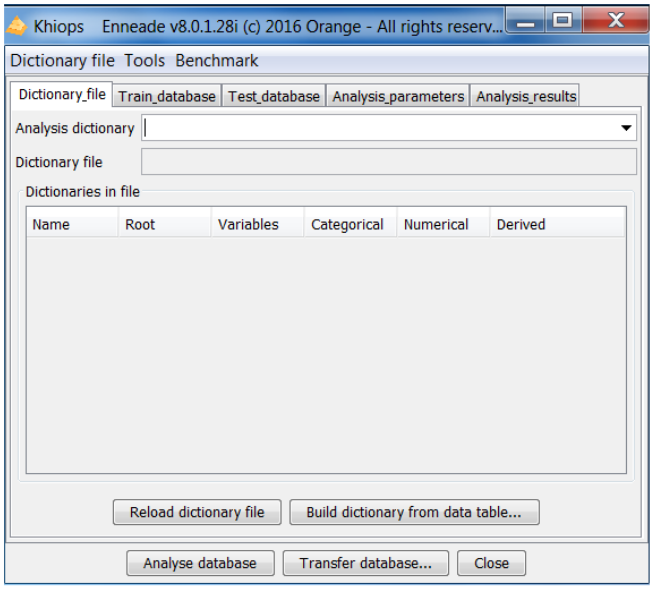


FIGURE 6.3

6.2 Clustering prédictif du premier type

Le clustering prédictif du premier type englobe l'ensemble des algorithmes du clustering modifiés permettant de prédire correctement la classe des nouvelles instances sous la contrainte d'avoir un nombre minimal de clusters. Dans ce cadre d'étude, l'axe de prédiction est principalement privilégié. L'algorithme des K-moyennes prédictives du premier type proposé dans cette thèse est donc l'algorithme incorporant les méthodes de prétraitement et d'initialisation des centres les plus performants en termes de prédictions.

Entrée
- Un ensemble de données D, où chaque instance X_i est décrite par un vecteur de d dimensions et par une classe $Y_i \in \{1, \ldots, J\}$.
- Le nombre de clusters souhaité, noté K.

Début
1) Prétraitement des données : *Conditional Info (CI-CI)*
2) Initialisation des centres : *Rocchio-And-Split*

Répéter
3) *Affectation :* générer une nouvelle partition en assignant chaque instance X_i au groupe dont le centre est le plus proche.

$$X_i \in C_k \forall j \in 1, \ldots, K \qquad k = argmin_j \parallel X_i - \mu_j \parallel$$

avec μ_k est le centre de gravité du cluster C_k.
4) *Représentation :* calculer les centres associés à la nouvelle partition

$$\mu_k = \frac{1}{N_k} \sum_{X_i \in C_k} X_i$$

jusqu'à ce que (convergence de l'algorithme)

5) Attribution des classes aux clusters formés : *vote majoritaire*
6) Prédiction de la classe des nouvelles instances : *un plus proche voisin*

Fin

Algorithme 8 – Algorithme des K-moyennes prédictives du premier type pour le cas CI-RS

En s'appuyant sur les résultats présentés dans la figure 6.1, l'algorithme des K-moyennes prédictives du premier type proposé est l'algorithme intégrant la méthode supervisée du prétraitement des données Conditional Info (CI) et la méthode supervisée d'initialisation des centres Rocchio-And-Split (RS). Pour un nombre fixe de clusters (K), l'algorithme 8 présente sous forme des lignes de code l'algorithme des K-moyennes prédictives du premier type.

L'un des principaux avantages de cet algorithme est qu'il n'est exécuté qu'une seule fois en raison de l'utilisation d'une méthode d'initialisation déterministe (*i.e.,* la méthode Rocchio-And-Split).

Cette section est consacrée à la comparaison des performances prédictives de cet algorithme des K-moyennes prédictives avec celles d'autres algorithmes du clustering prédictif les plus répandus dans la littérature. Cette section expérimentale est divisée en deux grandes parties. Dans la première partie (Section 6.2.1), on considère le nombre de clusters (K) comme une entrée de l'algorithme. Pour chaque jeu de données, on considère que le nombre de clusters (K) est égal au nombre de classes (J). Dans ce cas, le problème du départ devient un problème de classification supervisée. L'objectif de cette première partie est de tester la capacité de l'algorithme des K-moyennes prédictives présenté ci-dessus à atteindre l'objectif des algorithmes de la classification supervisée (*i.e.,* prédire correctement la classe des nouvelles instances).

La deuxième partie (Section 6.2.2) considère le nombre de clusters comme une sortie de l'algorithme (K≥J). L'objectif de cette partie est de tester la capacité de l'algorithme des K-moyennes prédictives à atteindre l'objectif des algorithmes de clustering prédictif du premier

type (*i.e.*, prédire correctement la classe des nouvelles instances sous contrainte d'obtenir un nombre minimal de clusters).

Les jeux de données utilisés dans cette partie expérimentale sont des jeux de données de l'UCI. Le tableau 6.1 présente les caractéristiques de ces jeux de données. Ces derniers ont été choisis afin d'avoir des bases de données illustratives diverses dans ce chapitre de synthèse en termes de nombre de classes J, de variables (continues M_n et/ou catégorielles M_c) et d'instances N. Pour chacun de ces jeux de données, on effectue un 2×5 folds cross validation.

ID	**Nom**	M_n	M_c	**N**	**J**	J_{maj}
1	Glass	10	0	214	6	36
2	Soybean	0	35	376	19	14
2	Breast	9	0	683	2	65
2	LED	7	0	1000	10	11
5	German	24	0	1000	2	70
6	Mushroom	0	22	8416	2	53

TABLE 6.1 – Liste des jeux de données utilisés- (J_{maj} représente $\approx$ pourcentage classe majoritaire).

6.2.1 Le nombre de clusters (K) est une entrée

Dans cette partie expérimentale, on cherche à tester la capacité de l'algorithme des K-moyennes prédictives présenté dans l'algorithme 8 à atteindre l'objectif des algorithmes de la classification supervisée. Les performances prédictives de l'algorithme des K-moyennes prédictives seront d'une part comparées à celles de l'algorithme des K-moyennes standard. Cette comparaison nous permet de savoir à quel point la version modifiée parvient à dépasser la version originale dans le contexte de la classification supervisée. D'autre part, l'algorithme des K-moyennes prédictives sera comparé à un des algorithmes de la classification supervisée le plus interprétable et le plus répandu dans la littérature, à savoir l'arbre de décision. Ce dernier est considéré comme une hiérarchie de clusters où chaque feuille représente un cluster. Pour une comparaison cohérente, le nombre de feuilles généré par l'arbre de décision est contrôlé de telle sorte d'avoir un nombre égal au nombre de classes du jeu de données utilisé (la taille du modèle est fixé K = J). Pour évaluer la performance prédictive de ces trois algorithmes, le critère "Variation d'Information" (VI) est utilisé. Plus la valeur de VI est proche de 0, meilleure est la performance prédictive du modèle.

Les deux figures 6.4 et 6.5 présentent les performances prédictives (en termes de VI) des trois algorithmes d'apprentissage lorsque le nombre de clusters (K) est égal au nombre de classes (J). Les résultats des deux figures montrent que l'algorithme des K-moyennes prédictives parvient à atteindre soit de meilleures performances prédictives par rapport à l'arbre de décision (résultats de la figure 6.4) ou des performances compétitives avec celles de l'arbre de décision (résultats de la figure 6.5). De plus, l'algorithme des K-moyennes prédictive arrive à atteindre des performances prédictives significativement meilleures que celles obtenues par l'algorithme des K-moyennes standard sachant que ce dernier est exécuté 100 fois avec différentes initialisations (en utilisant la même méthode K++) pour choisir la meilleure partition.

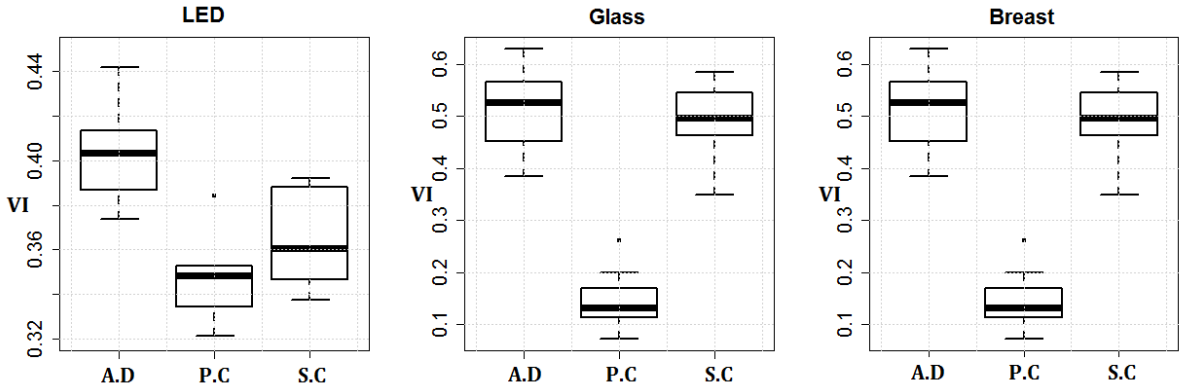


FIGURE 6.4 – Comparaison des performances prédictives (en termes de VI) pour les trois méthodes d'apprentissage (A.D = Arbre de décision, P.C= Prédictif clustering (K-moyennes prédictives) et S.C = K-moyennes standard)

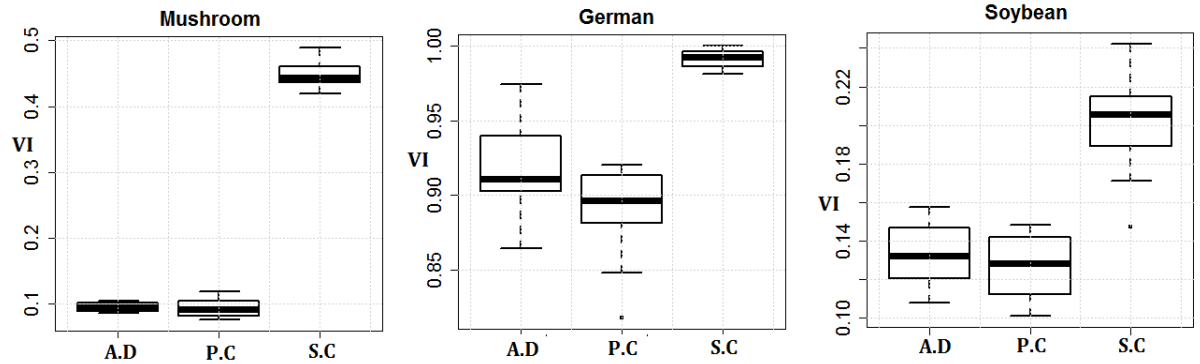


FIGURE 6.5 – Comparaison des performances prédictives (en termes de VI) pour les trois méthodes d'apprentissage (A.D = Arbre de décision, P.C= Prédictif clustering (K-moyennes prédictives) et S.C = K-moyennes standard)

6.2.2 Le nombre de clusters (K) est une sortie

Dans cette partie expérimentale, on cherche à tester la capacité de l'algorithme des K-moyennes prédictives présenté par l'algorithme 8 à atteindre l'objectif du clustering prédictif du premier type. Il s'agit ici de prédire correctement la classe des nouvelles instances sous la contrainte d'obtenir un nombre minimal de clusters (K).

Les algorithmes utilisés dans cette comparaison sont :

1. *L'arbre de décision* (A.D) : l'algorithme utilisé est l'arbre de décision de type CART. Dans cette étude expérimentale, nous utilisons l'algorithme élagué existant dans le logiciel R dans la librairie *rpart* [98]. Le nombre de clusters dans ce cas correspond au nombre de feuilles généré dans la phase d'apprentissage.
2. *Algorithme de Eick* (Eick) : cet algorithme nécessite un paramètre utilisateur β qui permet d'équilibrer le critère permettant d'évaluer la pureté des clusters en termes de classes et la contrainte sur le nombre de clusters générés. Les résultats présentés dans cette étude expérimentale sont obtenus lorsque β est égal à 0.1. Dans [46], Eick et al. montrent qu'avec cette valeur, ils parviennent à obtenir de meilleures performances. Pour plus de détails sur cet algorithme, voir la section 2.5.2 du chapitre 2.

3. *L'arbre du clustering prédictif* (PCT) : l'algorithme utilisé dans cette partie expérimentale est l'algorithme présenté dans "http ://clus.sourceforge.net/doku.php". Les résultats présentés dans cette section sont obtenus en utilisant l'arbre élagué présenté par cet algorithme. Dans ce cas, le nombre de clusters (K) correspond au nombre de feuilles générées par celui-ci dans la phase d'apprentissage. Pour plus de détails sur cet algorithme, voir la section 2.5.2 du chapitre 2.
4. *Les K-moyennes prédictives* (P.C) : pour avoir le nombre de clusters comme une sortie, l'algorithme des K-moyennes prédictives présenté par l'algorithme 8, est exécuté plusieurs fois avec différents nombres de clusters K dans le but de sélectionner la partition optimale au sens du clustering prédictif du premier type. Cette sélection est effectuée à l'aide de l'indice de rand ajusté "ARI".
5. *Les K-moyennes standard* (S.C) : pour avoir le nombre de clusters comme une sortie, l'algorithme des K-moyennes standard est exécuté plusieurs fois avec différent nombres de clusters K dans le but de sélectionner la partition optimale au sens du clustering prédictif du premier type. Cette sélection est effectuée à l'aide de l'indice de rand ajusté "ARI".

Les deux figures 6.6 et 6.7 présentent les performances prédictives (en termes de VI) des cinq modèles d'apprentissage. Les résultats de ces figures montrent qu'avec la supervision des deux étapes de prétraitement des données et d'initialisation des centres, l'algorithme des K-moyennes prédictives parvient à être très compétitif avec l'arbre de clustering prédictif (PCT). Cependant, l'algorithme des K-moyennes prédictives parvient à obtenir un nombre plus faible de clusters par rapport aux autres algorithmes. À titre d'exemple, pour le jeu de données German présenté dans la partie milieu de la figure 6.6, l'algorithme des K-moyennes prédictives obtient deux à trois clusters. Par contre, les arbres de décision, PCT, l'algorithme de Eick et les K-moyennes standard obtiennent respectivement 6, $\{63, 75\}$, $\{5-9\}$, 9 clusters.

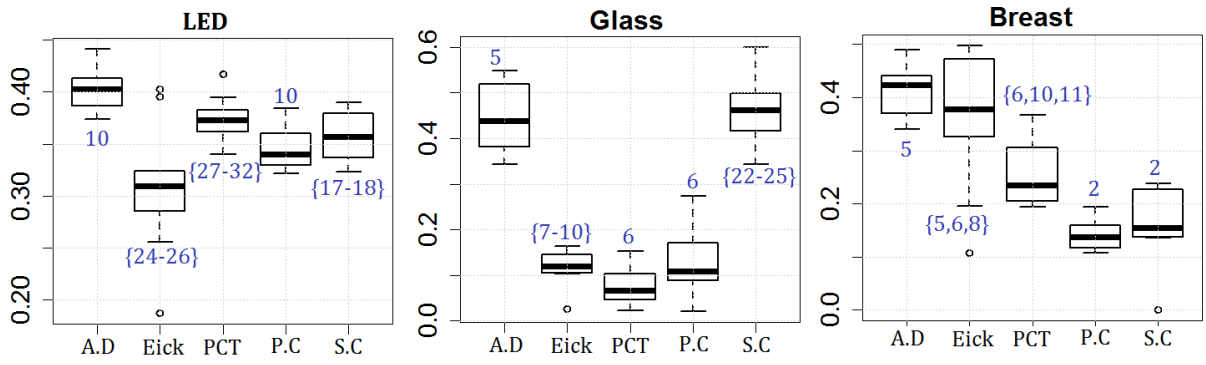


FIGURE 6.6 – Comparaison des performances prédictives (en termes de VI) des modèles lorsque le nombre de clusters est une sortie

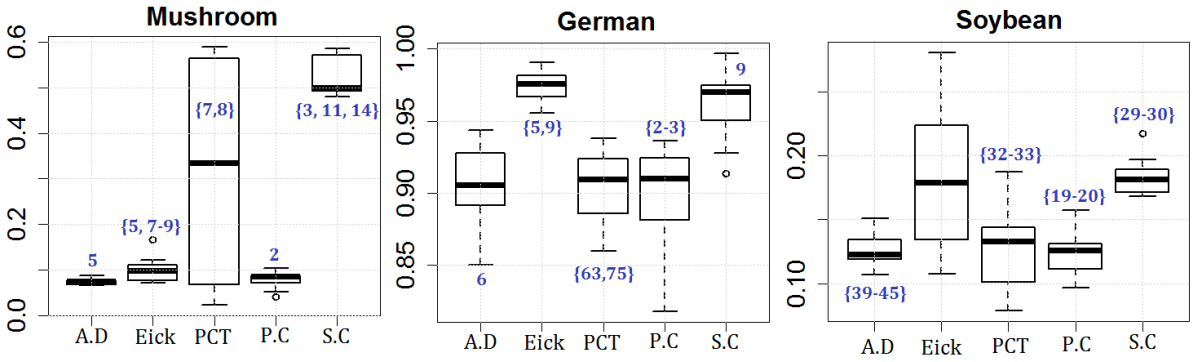


FIGURE 6.7 – Comparaison des performances prédictives (en termes de VI) des modèles lorsque le nombre de clusters est une sortie (suite)

6.2.3 Discussion

Dans cette première partie expérimentale, nous avons pu montrer qu'avec la supervision des deux étapes (prétraitement des données et initialisation des centres) de l'algorithme des K-moyennes standard, cet algorithme parvient à atteindre des performances (en termes de prédiction) meilleures ou très compétitives de celles obtenues par les algorithmes les plus répandus dans la littérature (*e.g.*, les arbres de décision et les arbres du clustering prédictif). Cet algorithme parvient également à avoir un nombre restreint de clusters par rapport à ces deux algorithmes. De plus, dans le contexte du clustering prédictif du premier type, on a pu montrer que l'algorithme des K-moyennes prédictives proposé est significativement meilleur que l'algorithme des K-moyennes standard.

6.3 Clustering prédictif du deuxième type

Le clustering prédictif du deuxième type englobe l'ensemble des algorithmes permettant de décrire et de prédire d'une manière simultanée. Il s'agit ici d'établir un compromis entre la description et la prédiction sous la contrainte d'interprétation des résultats. En s'appuyant sur la synthèse présentée dans 6.1 (voir la figure 6.2) et la discussion dans le chapitre d'initialisation (chapitre 4), l'algorithme des K-moyennes prédictives du deuxième type proposé dans cette thèse est l'algorithme des K-moyennes précédé par le prétraitement supervisé Conditional Info (CI) et la méthode d'initialisation KMeans++R (K++R). Pour le choix de la meilleure partition au sens du clustering prédictif du deuxième type, la version supervisée du critère Davies-Bouldin (SDB) est utilisée. Pour plus de détails sur ce critère, voir la section 5.3 du chapitre 5. L'algorithme 9 présente sous forme des lignes de code l'algorithme des K-moyennes prédictives du deuxième type.

Entrée
- Un ensemble de données D, où chaque instance X_i est décrite par un vecteur de d dimensions et par une classe $Y_i \in \{1, \ldots, J\}$.
- Le nombre de clusters souhaité, noté K.

Début
1) Prétraitement des données :*Conditional Info (CI-CI)*
2) Initialisation des centres : *Kmeans++R*
Pour K allant de J jusqu'à K_{max} **faire**
Répéter
3) *Affectation :* générer une nouvelle partition en assignant chaque instance X_i au groupe dont le centre est le plus proche.

$$X_i \in C_k \forall j \in 1, \ldots, K \qquad k = argmin_j \parallel X_i - \mu_j \parallel$$

avec μ_k est le centre de gravité du cluster C_k.
4) *Représentation :* calculer les centres associés à la nouvelle partition

$$\mu_k = \frac{1}{N_k} \sum_{X_i \in C_k} X_i$$

jusqu'à ce que (convergence de l'algorithme)
Fin Pour
5) Sélection du nombre de clusters optimal, noté K_{opti} en utilisant SDB.
6) Attribution des classes aux clusters formés : *vote majoritaire*
7) Prédiction de la classe des nouvelles instances. *un plus proche voisin.*
Fin

Algorithme 9 – Algorithme des K-moyennes prédictives du deuxième type pour le cas CI et K++R

L'objectif de cette section est donc de tester la capacité de cet algorithme à découvrir la structure interne *complète* de la variable cible. Il s'agit ici de découvrir les différentes raisons qui peuvent mener à une même prédiction. Cette partie expérimentale sert également à tester la capacité de cet algorithme à fournir des résultats facilement interprétables par l'utilisateur.

6.3.1 Description du jeu de données

Pour atteindre l'objectif cité ci-dessus, nous allons utiliser le jeu de données German de l'UCI. Ce jeu de données estime le risque crédit pour un demandeur donné. German contient 1000 demandeurs de crédit. Chaque demandeur est décrit par 20 variables descriptives et un score décrivant son risque crédit. 700 demandeurs ont été qualifiés à risque faible (classe 1) et 300 à risque élevé (classe 2). Une description détaillée de ce jeu de données est présentée dans "http ://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)".

La description des 20 variables descriptives est donnée dans ce qui suit :

1. Attribute 1 (V1) : Status of existing checking account (qualitative)
   — A11 : ... < 0 DM
   — A12 : 0 <= ... < 200 DM

- A13 : ... >= 200 DM
- A14 : no checking account

2. Attribute 2 (V2) : Duration in month (numerical)
3. Attribute 3 (V3) : Credit history (qualitative)
   - A30 : no credits taken/all credits paid back duly
   - A31 : all credits at this bank paid back duly
   - A32 : existing credits paid back duly till now
   - A33 : delay in paying off in the past
   - A34 : critical account/other credits existing (not at this bank)
4. Attribute 4 (V4) : Purpose (qualitative)
   - A40 : car (new)
   - A41 : car (used)
   - A42 : furniture/equipment
   - A43 : radio/television
   - A44 : domestic appliances
   - A45 : repairs
   - A46 : education
   - A47 : (vacation - does not exist ?)
   - A48 : retraining
   - A49 : business
   - A410 : others
5. Attribute 5 (V5) : Credit amount (numerical)
6. Attibute 6 (V6) : Savings account/bonds (qualitative)
   - A61 : ... < 100 DM
   - A62 : 100 <= ... < 500 DM
   - A63 : 500 <= ... < 1000 DM
   - A64 : .. >= 1000 DM
   - A65 : unknown/ no savings account
7. Attribute 7 (V7) : Present employment since (qualitative)
   - A71 : unemployed
   - A72 : ... < 1 year
   - A73 : 1 <= ... < 4 years
   - A74 : 4 <= ... < 7 years
   - A75 : .. >= 7 years
8. Attribute 8 (V8) : Installment rate in percentage of disposable income (numerical)
9. Attribute 9 (V9) : Personal status and sex (qualitative)
   - A91 : male : divorced/separated
   - A92 : female : divorced/separated/married
   - A93 : male : single
   - A94 : male : married/widowed
   - A95 : female : single
10. Attribute 10 (V10) : Other debtors / guarantors (qualitative)
    - A101 : none
    - A102 : co-applicant
    - A103 : guarantor

11. Attribute 11 (V11) : Present residence since (numerical)
12. Attribute 12 (V12) : Property (qualitative)
    — A121 : real estate
    — A122 : if not A121 : building society savings agreement/life insurance
    — A123 : if not A121/A122 : car or other, not in attribute 6
    — A124 : unknown / no property
13. Attribute 13 (V13) : Age in years (numerical)
14. Attribute 14 (V14) : Other installment plans (qualitative)
    — A141 : bank
    — A142 : stores
    — A143 : none
15. Attribute 15 (V15) : Housing (qualitative)
    — A151 : rent
    — A152 : own
    — A153 : for free
16. Attribute 16 (V16) : Number of existing credits at this bank (numerical)
17. Attribute 17 (V17) : Job (qualitative)
    — A171 : unemployed/ unskilled - non-resident
    — A172 : unskilled - resident
    — A173 : skilled employee / official
    — A174 : management/ self-employed/highly qualified employee/ officer
18. Attribute 18 (V18) : Number of people being liable to provide maintenance for (numerical)
19. Attribute 19 (V19) : Telephone (qualitative)
    — A191 : none
    — A192 : yes, registered under the customers name
20. Attribute 20 (V20) : foreign worker (qualitative)
    — A201 : yes
    — A202 : no

6.3.2 Résultats

Pour le jeu de données présenté ci-dessus, on cherche à connaître les différentes raisons qui peuvent mener à un faible ou à un fort risque crédit. Pour ce faire, l'algorithme des K-moyennes présenté dans l'algorithme 9, est exécuté avec différent nombres de clusters (voir la boucle **Pour** de l'algorithme 9) dans le but de sélectionner la partition optimale en utilisant le critère SDB. Il est à rappeler que la partition optimale au sens du clustering prédictif du deuxième type est celle qui réalise un bon compromis entre la description et la prédiction.

La figure 6.8 présente la valeur du critère SDB pour différentes partitions générées par l'algorithme des K-moyennes prédictives ayant différents nombres de clusters ($K \in [2, 14]$). Cette figure montre que la partition optimale au sens du clustering prédictif du deuxième type est la partition ayant 7 clusters.

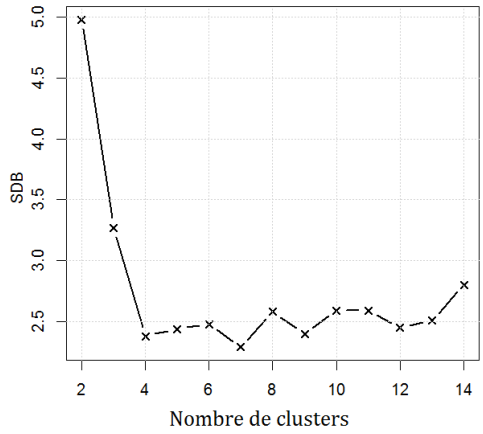


FIGURE 6.8 – Nombre de clusters K versus SDB

Cette partition optimale contient deux différents clusters formés par des demandeurs à fort risque crédit (clusters 2 et 4) et 5 différents clusters (clusters 1, 3, 5, 6 et 7) formés par des demandeurs à faible risque crédit. Le tableau 6.2 présente la probabilité que les demandeurs d'un cluster appartiennent à une des classes. Pour la première classe, c'est-à-dire, la classe à faible risque, les résultats du tableau 6.2 montrent que les trois clusters 1, 5 et 6 sont les clusters les plus purs en termes de la classe 1.

	Classe 1	Classe 2	# instances
Cluster 1	0.84	0.16	273
Cluster 5	0.82	0.18	91
Cluster 6	0.80	0.21	200
Cluster 3	0.65	0.35	193
Cluster 7	0.54	0.46	46
Cluster 2	0.48	0.52	154
Cluster 4	0.26	0.74	43

TABLE 6.2 – La probabilité d'appartenir à une classe j ($j \in \{1, 2\}$) conditionnellement au cluster i ($i \in \{1, \ldots, 7\}$)

Bien que les demandeurs qui forment ces 3 clusters ont la même étiquette, ils ont des profils différents. En effet, la figure 6.9 présente la description des différents clusters en utilisant les variables les plus discriminantes (V1, V2, V3, V4, V6, V12 et V15). Cette figure présente, pour chaque cluster et pour chacune des variables, les quantités d'informations (obtenues grâce au prétraitement Conditional Info) existant dans chaque intervalle de discrétisation ou groupe de modalités. Chaque bâton de cette figure présente la probabilité d'appartenir à un intervalle de discrétisation ou à un groupe de modalités (selon la nature de la variable en question) conditionnellement aux clusters. Cette figure montre que :

— Les gens du **cluster 1** sont spécifiquement des gens ayant une grand somme dans leur compte épargne (voir la variable V6).
— Les gens du **cluster 5** sont spécifiquement des gens ayant une grand somme dans leur compte courant (voir la variable V1) et ils possèdent des biens tels que des voitures, une assurance de vie, *etc.* (voir la variable V12).

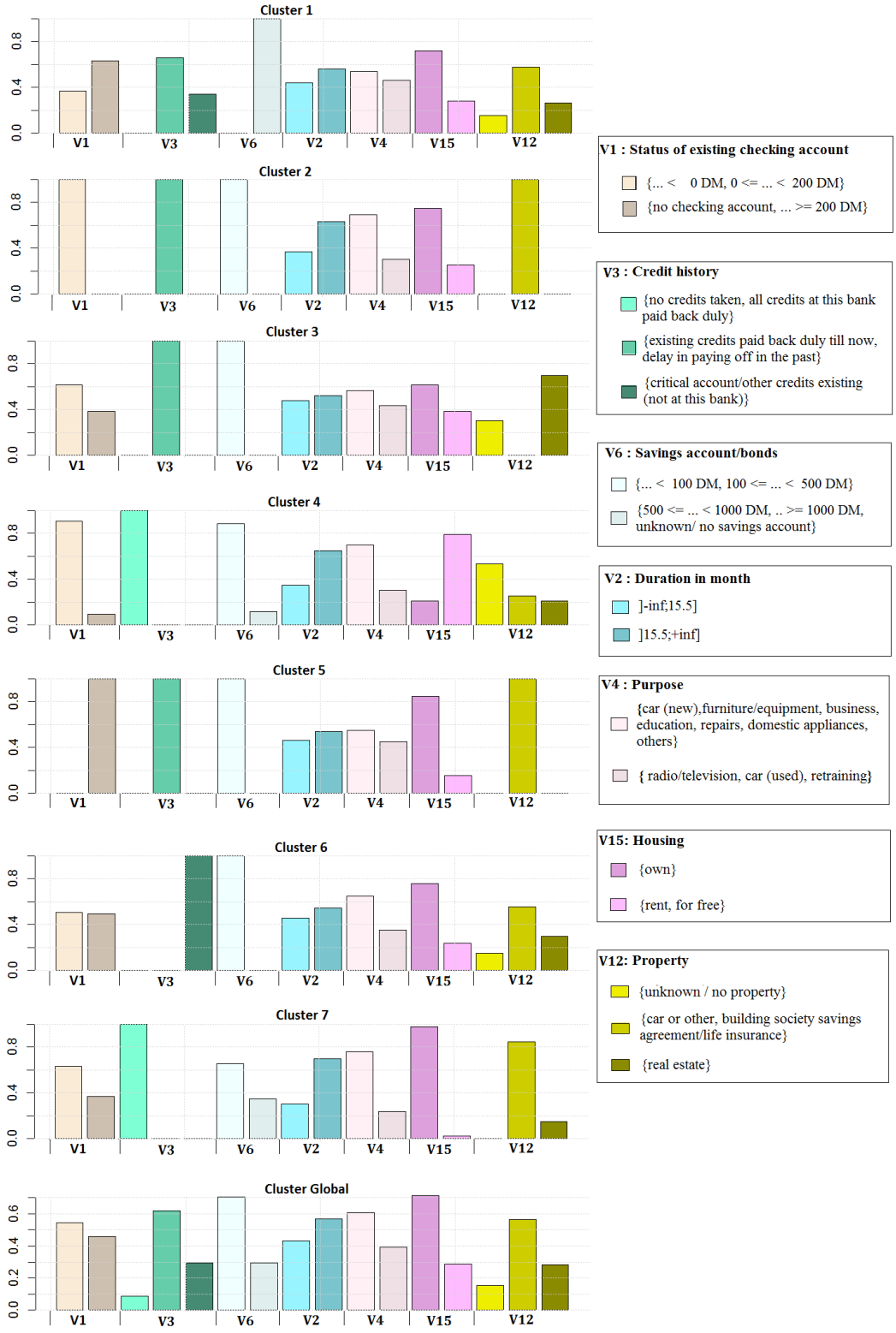


FIGURE 6.9 – Description des clusters de la partition optimale

— Les gens du **cluster 6** sont de gens ayant un autre crédit dans une autre banque (voir la variable V3). Cependant, la majorité de ces gens possèdent des biens immobiliers ou une assurance de vie, des voitures, etc (voir la variable V12). Ils ont également leur propre maison (voir la variable V15).

Bien que les deux clusters 3 et 7 sont les moins purs en termes de la classe 1, mais ils fournissent des informations différentes des trois premiers clusters et qui semblent être pertinentes :

— **Le cluster** 3 contient des gens qui ont déjà un crédit dans cette banque. Les dates du remboursement des parts de ce crédit ont été respectées jusqu'à présent. La majorité de ces gens possèdent des biens immobiliers.

— **Le cluster 7** contient spécifiquement des gens qui n'ont pas eu de crédits auparavant ou bien tous les crédits déjà pris par ces gens ont été remboursés. Ces gens possèdent soit des biens immobiliers, soit des assurances de vie ou des voitures, etc.

En ce qui concerne la deuxième classe, c'est-à-dire, la classe ayant des gens à fort risque crédit, on trouve que celle-ci peut être partitionnée en deux différents sous-groupes (cluster 2 et cluster 4).

— **Le cluster 2** contient spécifiquement des gens qui ont une petite somme dans leur compte courant et dans leur compte d'épargne. Ils ont déjà un crédit en cours dans cette banque ou les crédits qu'ils ont eu dans le passé n'ont pas été remboursés dans leurs délais. Ces gens possèdent des voitures, des assurances de vie, etc.

— **Le cluster 4** contient des gens qui ont une somme entre 0 et 200 DM dans leur compte courant. Ces gens n'ont pas eu de crédits auparavant ou bien tous les crédits déjà pris par ces gens dans cette banque ont été remboursés. Cependant, la majorité de ces gens n'ont pas de propriétés. Ils louent une maison ou ils sont hébergés gratuitement chez quelqu'un de proche.

6.3.3 Discussion

À l'aide de cet exemple illustratif, nous avons pu montrer la capacité de l'algorithme des K-moyennes prédictives proposé dans cette thèse à atteindre l'objectif du clustering prédictif du deuxième type et donc découvrir les différentes raisons qui peuvent mener à une même prédiction. Pour le jeu de données German, nous avons pu montrer les différents profils des demandeurs de crédits existant dans la base de données. De plus, nous avons montré la capacité de cet algorithme à fournir des résultats facilement interprétables par l'utilisateur.

6.4 Conclusion & Perspectives

6.4.1 Conclusion générale

Le clustering prédictif est un cadre dans lequel décrire et prédire d'une manière simultanée est un nouvel aspect d'apprentissage supervisé qui englobe à la fois les caractéristiques de la classification supervisée et du clustering. Afin de mettre en évidence les difficultés liés à ce type d'apprentissage, nous avons commencé dans cette thèse par introduire les concepts clés du clustering et de la classification supervisée. Nous avons ensuite présenté une liste non exhaustive des différentes approches potentielles permettant d'atteindre l'objectif souhaité.

Pour atteindre l'objectif de la thèse qui est la recherche d'un modèle "interprétable" permettant de décrire et de prédire d'une manière simultanée, nous avons choisi de modifier l'algorithme des K-moyennes standard. Cette version modifiée contient 7 étapes dont chacune peut être supervisée indépendamment des autres. Dans cette thèse, nous nous sommes intéressés à la supervision

de quatre étapes de cet algorithme, à savoir : 1) le prétraitement des données, 2) l'initialisation des centres, 3) le choix de la meilleure partition et 4) la mesure d'importance des variables.

Pour l'étape du prétraitement supervisé, nous avons pu montrer dans le chapitre 3 que les deux prétraitements proposés "Conditional Info" (CI) et Binarization (BIN) ont la capacité d'aider l'algorithme des K-moyennes standard à atteindre l'objectif du clustering prédictif. En effet, CI et BIN permettent d'écrire une distance dépendante de la classe permettant d'établir une relation entre la proximité des instances en termes de distance et leur classe d'appartenance. Ces méthodes de prétraitement sont donc capables de modifier indirectement la fonction du coût de l'algorithme des K-moyennes standard dans le but de l'adapter au problème du clustering prédictif. Les expérimentations menées dans le chapitre 3 ont montré que : *i*) lorsque l'axe du prédiction est privilégié, l'algorithme des K-moyennes standard précédé par Conditional Info fournit des résultats significativement meilleurs que l'algorithme des K-moyennes standard (précédé par les prétraitements non supervisés), *ii*) lorsque l'axe de description est privilégié, les prétraitements supervisés (Conditional Info et Binarization) parviennent à construire de bonnes matrices de Gram relativement à la variable cible. Nous avons pu montrer également que ces méthodes de prétraitement supervisées aident l'algorithme des K-moyennes standard à fournir des résultats facilement interprétables par l'utilisateur.

Pour l'étape d'initialisation des centres, nous avons présenté dans le chapitre 4 l'influence de l'utilisation d'une méthode d'initialisation supervisée ou non supervisée sur la qualité (au sens du clustering prédictif) des résultats générés par l'algorithme des K-moyennes standard. Dans ce cadre d'étude, nous avons proposé trois méthodes supervisées d'initialisation des centres (une déterministe et deux basées sur l'aléatoire) : Rocchio-And-Split (RS), KMeans++R et S-Bisecting. À l'aide de ces méthodes, nous avons pu montrer qu'une bonne méthode supervisée d'initialisation a la capacité d'aider l'algorithme des K-moyennes standard à atteindre l'objectif du clustering prédictif. Lorsque l'axe de prédiction est privilégié (le cas du clustering prédictif du premier type), nous avons pu montrer que quel que soit le prétraitement utilisé (supervisé ou non supervisé), l'algorithme des K-moyennes standard précédé par la méthode Rocchio-And-Split (RS) fournit de meilleures performances prédictives par rapport aux autres méthodes. Il est important de signaler qu'en raison de l'utilisation de RS (méthode déterministe), l'algorithme des K-moyennes n'est exécuté qu'une seule fois. Lorsqu'on cherche à réaliser le compromis entre la description et la prédiction (cas du clustering prédictif du deuxième type), nous avons pu montrer que quel que soit le prétraitement utilisé, la méthode S-Bisecting (SB) et la méthode K++R sont celles qui aident l'algorithme des K-moyennes standard à obtenir un bon compromis entre la description et la prédiction par rapport aux autres méthodes.

Pour l'étape du choix de la meilleure partition, nous avons commencé par présenter dans le chapitre 5 l'influence de l'utilisation d'un critère supervisé ou non supervisé sur la qualité des résultats obtenus par l'algorithme des K-moyennes au sens du clustering prédictif du premier type. Les résultats expérimentaux ont montré que l'utilisation de l'indice de rand ajusté (ARI) pour choisir la meilleure partition permet à l'algorithme des K-moyennes standard de gagner sur l'axe de prédiction. En raison d'absence d'un critère analytique permettant de mesurer la qualité des résultats issus des algorithmes du clustering prédictif du deuxième type, nous avons proposé dans le chapitre 5 une version supervisée de l'indice de Davies-Bouldin, nommée SDB. Cet indice est basé sur une nouvelle mesure de dissimilarité capable d'établir une relation entre la proximité des instances en termes de distance et leur classe d'appartenance : deux instances sont considérées comme similaires suivant cette nouvelle mesure si et seulement si elles sont proches en termes de distance et ont également la même étiquette. Grâce à cette version supervisée de l'indice de Davies-Bouldin, nous avons pu surmonter le problème de la non corrélation entre les clusters et les classes. Dans ce cadre, les résultats expérimentaux ont montré que l'indice SDB

parvient bien à mesurer le compromis entre la description et la prédiction.

Pour l'étape de la mesure d'importance des variables, nous avons proposé dans l'annexe E une méthode supervisée permettant de mesurer l'importance des variables après la convergence du modèle. L'importance d'une variable est mesurée dans cette étude par son pouvoir à prédire l'ID-clusters. Le problème dans ce cas devient un problème "uni-varié" de la classification supervisée. Il s'agit ici de remplacer la variable cible par l'ID-clusters obtenu par l'algorithme du clustering prédictif et ensuite de mesurer la capacité de chaque variable à prédire correctement l'ID-cluster en utilisant un des algorithmes de la classification supervisée. Dans cette étude, nous n'avons pas considéré les interactions qui peuvent exister entre les variables (e.g., une variable n'est importante qu'en présence des autres). Ceci fait l'objet des futurs travaux (parmi d'autre, voir la section 6.4.2).

Dans les chapitres précédents, chaque étape a été traitée individuellement. Dans le début de ce chapitre de synthèse, nous avons regroupé l'ensemble des méthodes supervisées proposées au cours de cette thèse (prétraitement, initialisation et le critère d'évaluation pour choisir la meilleure partition) dans un seul algorithme nommé, K-moyennes prédictives. Dans le cadre du clustering prédictif du premier type, les résultats expérimentaux ont montré qu'avec la supervision des deux étapes de prétraitement et d'initialisation, l'algorithme des K-moyennes parvient à être meilleur ou très compétitif avec les algorithmes de la littérature tel que l'arbre de décision et l'arbre de clustering prédictif. Dans le cadre du clustering prédictif du deuxième type, nous avons pu montrer que l'algorithme proposé dans cette thèse arrive à découvrir les différentes raisons qui peuvent mener à une même prédiction et donc découvrir la structure interne de la variable cible. Les résultats fournis par notre algorithme sont présenté sous forme d'histogrammes facilement interprétables par l'utilisateur.

6.4.2 Perspectives

Parmi les différents travaux futurs qui pourraient être envisagés suite à cette thèse, nous proposons les pistes suivantes :

1. **Modification de la fonction du coût de l'algorithme des K-moyennes** : l'algorithme des K-moyennes standard est basé sur une distance (souvent la distance Euclidienne) pour mesurer la proximité entre les instances. Cette distance n'accorde aucune importance à l'étiquetage des instances : deux instances de différentes étiquettes peuvent être considérées comme similaires si elles sont proches l'une de l'autre. Pour surmonter ce problème l'utilisation d'une mesure qui permet d'établir une relation entre la proximité des instances et leur classe d'appartenance s'avère nécessaire. Dans ce contexte, l'utilisation de la nouvelle mesure de dissimilarité proposée dans cette thèse dans le chapitre 5 pourrait résoudre ce problème.

2. **La prédiction de la classe des nouvelles instances** : durant cette thèse, la prédiction de la classe des nouvelles instances est effectuée à l'aide de l'approche un plus proche voisin : chaque nouvelle instance prend la classe du cluster qui lui est le plus proche. Dans certains cas (*e.g.,* le cas de déséquilibre des classes), l'utilisation du modèle un plus proche voisin s'avère insuffisant. En effet, il se peut que le cluster le plus proche de la nouvelle instance ait une étiquette différente de celle-ci. La solution la plus intuitive qui pourrait résoudre ce problème et d'améliorer d'avantage la qualité prédictive de l'algorithme des K-moyennes prédictives est l'utilisation du modèle k-plus proches voisins ($k > 2$).

3. **Amélioration des performances prédictives** : dans le cadre du clustering prédictif du premier type, l'axe de prédiction est privilégié. Pour améliorer davantage la performance prédictive de l'algorithme des K-moyennes prédictives du premier type, le modèle LVQ pourrait être utilisé. Ce dernier prendrait en entrée les centres générés par l'algorithme des K-moyennes prédictives après convergence. Ce modèle pourrait également être appliqué après l'étape d'initialisation des centres.

4. **Mesure d'importance des variables** : dans les travaux réalisé à ce sujet (voir l'annexe E), nous avons suivi une méthode uni-variée où chaque variable est traitée indépendamment des autres. Dans ce cas, nous n'avons pas étudié les interactions qui peuvent exister entre les variables. En effet, dans certains cas, une variable descriptives n'est importante qu'en présence d'une ou d'autres variables. Pour les travaux futurs, il est important de considérer donc ces interactions.

Liste des publications

Revue internationale

[10] Oumaima Alaoui Ismaili, Vincent Lemaire, Antoine Cornuéjols. **Supervised pre-processings are useful for supervised clustering**. In *Springer Series Studies in Classification, Data Analysis, and Knowledge Organization*, Bremen, 2015.

Conférences internationales

[13] Oumaima Alaoui Ismaili, Vincent Lemaire, Antoine Cornuéjols. **Evaluation of predictive clustering quality**, in MBC2, on Model Based clustering and classification (MBC2,2016).

[70] Vincent Lemaire, Oumaima Alaoui Ismaili, and Antoine Cornuéjols. **An initialization scheme for supervized k-means**. In *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland*, July 12-17, 2015, pages 1–8, 2015.

[59] Oumaima Alaoui Ismaili, Vincent Lemaire, Antoine Cornuéjols. **A Supervised Methodology to Measure the Variables Contribution to a Clustering**, in Neural Information Processing - 21st International Conference, (ICONIP 2014), Kuching, Malaysia. pp. 159–166, 2014.

Revue nationale

[12] Oumaima Alaoui Ismaili, Vincent Lemaire, Antoine Cornuéjols : **Clustering prédictif : décrire, prédire et interpréter simultanément** à venir sur invitation suite à la conférence RJCIA, in Revue d'intelligence Artificielle (RIA).

Conférences nationales

[60] Oumaima Alaoui Ismaili, Vincent Lemaire, and Antoine Cornuéjols. **Une méthode supervisée pour initialiser les centres des k-moyennes**. *Extraction et Gestion des Connaissances (EGC), Reims*, 2016.

[11] Oumaima Alaoui Ismaili, Vincent Lemaire, and Antoine Cornuéjols. **Une initialisation des K-moyennes à l'aide d'une décomposition supervisée des classes**. *Congrès de la Société Française de Classification (SFC)*, Nantes, 2015.

[9] Oumaima Alaoui Ismaili, Vincent Lemaire, Antoine Cornuéjols. **Classification à base de clustering ou comment décrire et prédire simultanément ?**. Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJCIA),Rennes, pages 7-12, 2015.

[8] Oumaima Alaoui Ismaili, Vincent Lemaire, Antoine Cornuéjols. **Une méthode basée sur des effectifs pour calculer la contribution des variables à un clustering**, In Atelier CluCo de la conférence Extraction et Gestion des Connaissances (EGC 2014), Rennes.

Démonstration

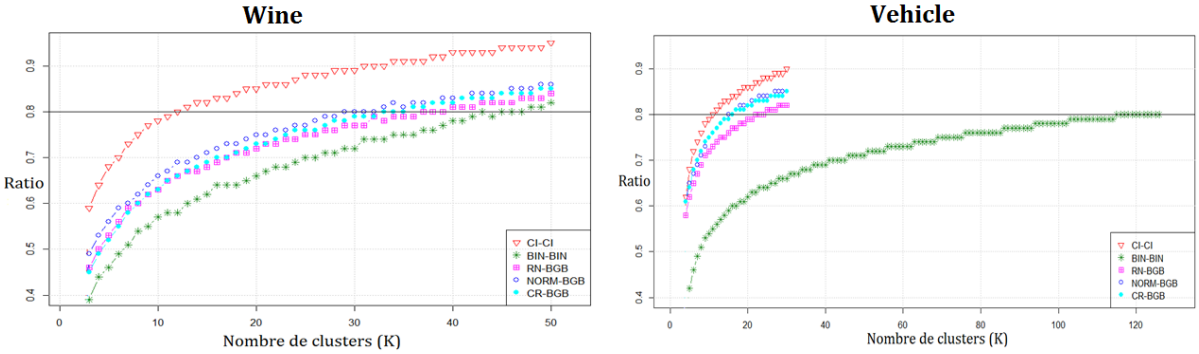
[69] Vincent Lemaire, Oumaima Alaoui Ismaili. **Un outil pour la classification à base de clustering pour décrire et prédire simultanément.** In Atelier Clustering and Co-clustering (CluCo), EGC 2016, Reims.

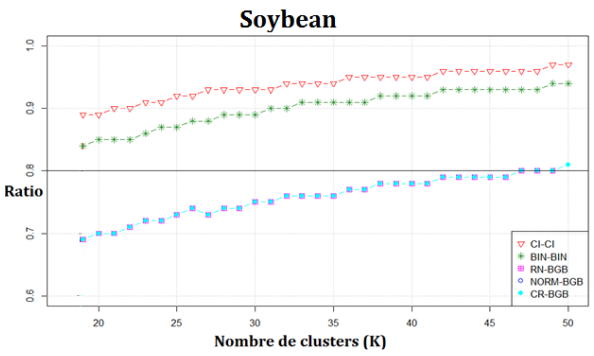
Annexes

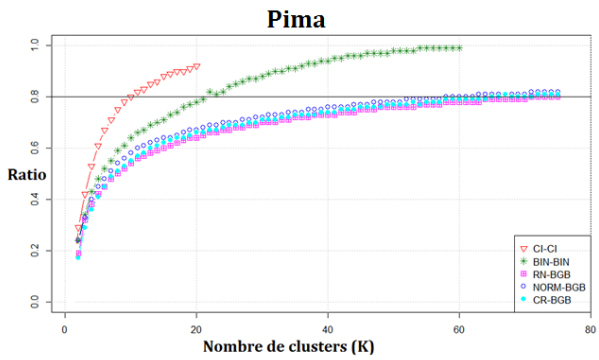
Annexe A

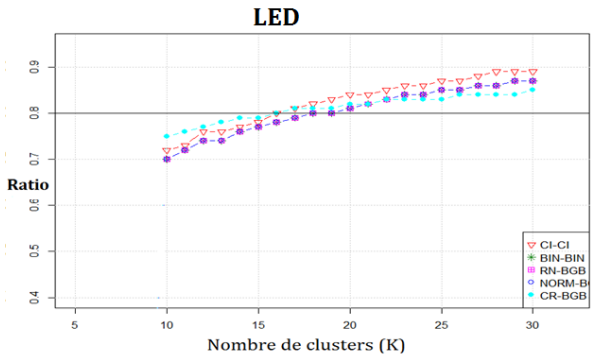
Plage de variation du nombre de clusters K

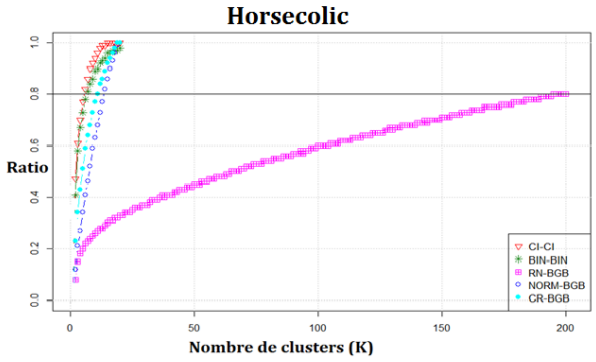
L'une des limites de l'algorithme des K-moyennes réside dans le fait que le nombre de centres est un paramètre utilisateur. Cela nécessite soit de connaître à l'avance le nombre de clusters que l'on souhaite obtenir, ce qui est rarement le cas, soit de découvrir des clusters naturels dans les données. Une façon de faire est d'exécuter l'algorithme un grand nombre de fois avec différentes valeurs de K et de choisir la meilleure segmentation. Dans notre cas d'étude, nous varions le nombre de clusters de C jusqu'à K_i pour un prétraitement i donné. Plusieurs techniques existent pour déterminer le nombre maximal de clusters K_i. Par exemple, on peut prendre pour chaque jeu de données, $K_i = \sqrt{N}$ avec N est le nombre d'instances dans l'ensemble des données. Dans ce qui suit, pour chaque jeu de données, K_i est déterminé au préalable de manière à ce que la partition obtenue, avec $K{=}K_i$ permette d'obtenir un ratio (inertie inter / inertie totale) de 80% ou de 90% si $K_i < C$. En procédant ainsi, le nombre de clusters maximal K_i déterminé dépendra du jeu de données et du prétraitement utilisés. Il est à rappeler que le nombre de cluster K doit être impérativement supérieur ou égal au nombre de classe puisqu'on souhaite découvrir la structure interne de la variable cible. La figure A.1 présente l'évolution du ratio (inertie inter / inertie) en fonction du nombre de clusters pour chaque prétraitement et pour chaque jeu de données.

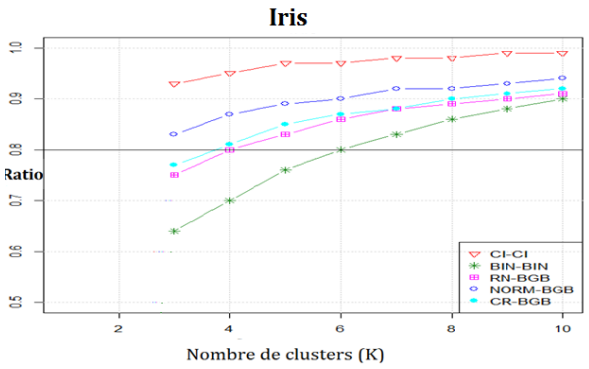


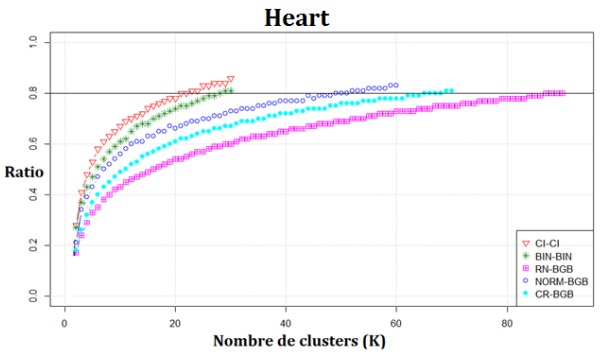

Soybean
Ratio
Nombre de clusters (K)
CI-CI
BIN-BIN
RN-BGB
NORM-BGB
CR-BGB

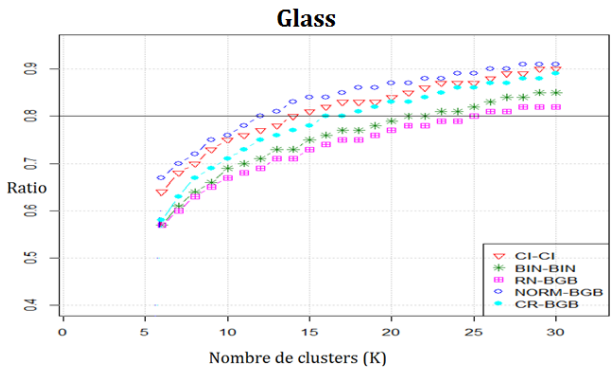

Pima
Ratio
Nombre de clusters (K)
CI-CI
BIN-BIN
RN-BGB
NORM-BGB
CR-BGB

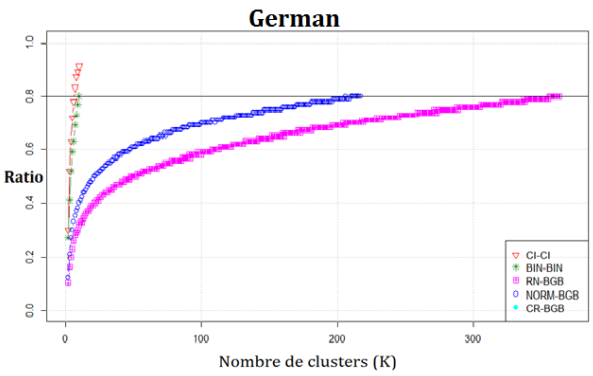

LED
Ratio
Nombre de clusters (K)
CI-CI
BIN-BIN
RN-BGB
NORM-B
CR-BGB


Horsecolic
Ratio
Nombre de clusters (K)
CI-CI
BIN-BIN
RN-BGB
NORM-BGB
CR-BGB


Iris
Ratio
Nombre de clusters (K)
CI-CI
BIN-BIN
RN-BGB
NORM-BGB
CR-BGB


Heart
Ratio
Nombre de clusters (K)
CI-CI
BIN-BIN
RN-BGB
NORM-BGB
CR-BGB


Glass
Ratio
Nombre de clusters (K)
CI-CI
BIN-BIN
RN-BGB
NORM-BGB
CR-BGB


German
Ratio
Nombre de clusters (K)
CI-CI
BIN-BIN
RN-BGB
NORM-BGB
CR-BGB

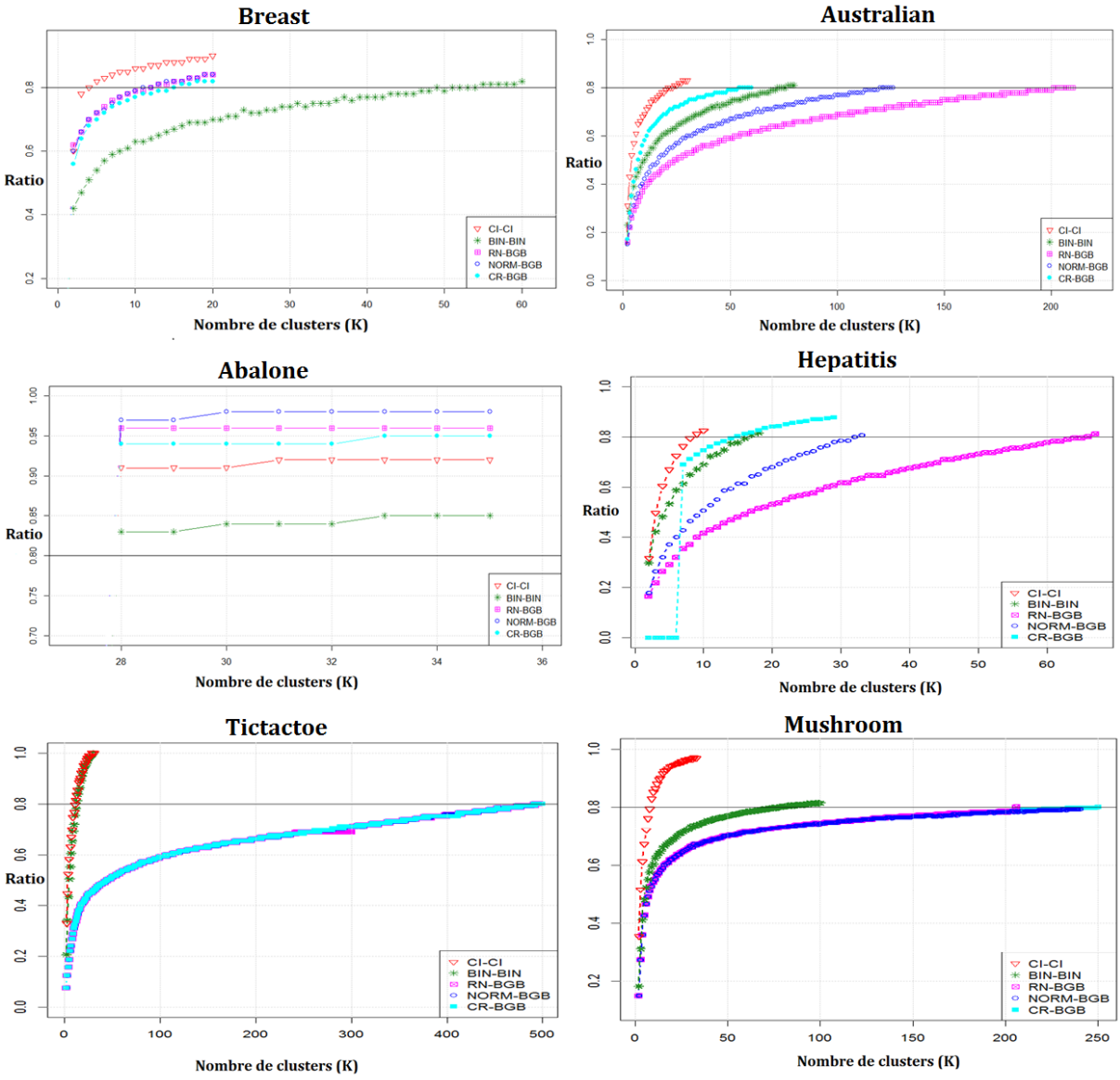


FIGURE A.1 – Evolution du Ratio (inertie inter / inertie) en fonction du nombre de clusters pour chaque prétraitement en prenant $R = 100$ et 100% en apprentissage.

Données	CI-CI	BIN-BIN	RN-BGB	CR-BGB	NORM-BGB
Iris	3	6	4	4	4
Glass	15	21	25	17	12
Wine	12	47	38	35	33
Vehicle	11	126	24	17	16
LED	17	19	19	17	19
Heart	23	29	90	67	53
Pima	10	22	74	69	61
Tictactoe	12	13	496	499	500
German	7	10	363	280	217
Australian	22	74	210	58	126
Abalone	29	29	29	29	29
Segmentation	23	64	21	15	8
Breast	4	56	12	15	12
Horsecolic	6	7	200	11	14
Hepatitis	9	17	66	28	32
PenDigits	73	64	64	28	22
Soybean	20	20	49	49	49
Mushroom	8	78	206	241	240
Waveform	86	64	64	64	64
Adult	12	64	64	214	64

TABLE A.1 – Détermination de K_i pour chaque prétraitement i

Annexe B

Chapitre 3

B.1 Le test de Friedman couplé au test post-hoc de Nemenyi

Le test de Friedman est une alternative non paramétrique à l'ANOVA à deux facteurs dans le cas où l'hypothèse de normalité n'est pas acceptable. Il permet de détecter les différences entre les groupes dans U échantillons appariés ($U > 2$) de taille n. Dans notre cas, chaque groupe représente les performances prédictives d'une méthode de prétraitement sur les N jeux de données. L'hypothèse nulle de ce test suppose que toutes les moyennes des groupes sont égales et l'hypothèse alternative qu'il existe au moins un couple (i, j) tel que la moyenne du groupe i est différente de la moyenne du groupe j. Le test est basé sur la somme des rangs de l'échantillon considéré. Soit n la taille des U échantillons appariés, la statistiques Q du test de Friedman est donnée par :

$$Q = \frac{12}{n(n+1)U} \sum_{i=1}^{U} [R_i^2 - 3n(U+1)] \tag{B.1}$$

où R_i est la somme des rangs pour l'échantillon i. Lorsqu'il y a des exæquo, les rangs moyens pour les observations correspondantes sont alors utilisés. La statistique Q est approximée par une loi du Khi Deux à $(U-1)$ degrés de liberté. Cette approximation est fiable lorsque $nU > 30$.

Le post-hoc de Nemenyi est une méthode de comparaison multiple qui permet de trouver les groupes qui diffèrent après qu'un test statistique de comparaison multiple (tel que le test de Friedman) ait rejeté l'hypothèse nulle. Le test de Nemenyi est similaire au test de Tukey pour l'ANOVA et il est utilisé quand tous les groupes sont comparables les uns aux autres. Dans notre cas d'étude, la performance des deux méthodes de prétraitement est significativement différente si les rangs moyens correspondants diffèrent d'au moins une différence critique, notée $CD = q_\alpha \sqrt{\frac{U(U+1)}{6Z}}$ où la valeur critique q_α est basé sur la statistique de Student divisé par $\sqrt{2}$.

Le tableau B.1 présente les différentes valeurs de q pour différent nombre de méthodes (Z) lorsque l'erreur du premier degré α est égale à 0.05 et à 0.10.

#Méthodes	2	3	4	5	6	7	8	9	10
$q_{0.05}$	1.960	2.343	2.569	2.728	2.850	2.949	3.031	3.102	3.16
$q_{0.10}$	1.645	2.052	2.291	2.459	2.589	2.693	2.780	2.855	2.920

TABLE B.1 – Les valeurs de q pour le test Nemenyi

B.2 Résultats : Prétraitements supervisés Vs. Prétraitement non supervisés

B.2.1 Cas où le nombre de clusters K est égal au nombre de classes J

Données	Méthodes	Apprentissage	Test	Données	Méthodes	Apprentissage	Test
Wine ($K=3$)	RN-BGB	$\mathbf{0.91 \pm 0.01}$	0.88 ± 0.01	Vehicle ($K=4$)	RN-BGB	0.08 ± 0.01	0.08 ± 0.01
	CR-BGB	$\mathbf{0.91 \pm 0.02}$	$\mathbf{0.86 \pm 0.02}$		CR-BGB	0.08 ± 0.01	0.08 ± 0.01
	NORM-BGB	0.85 ± 0.01	0.83 ± 0.01		NORM-BGB	0.08 ± 0	0.08 ± 0
	BIN-BIN	$\mathbf{0.91 \pm 0.02}$	$\mathbf{0.86 \pm 0.02}$		BIN-BIN	$\mathbf{0.18 \pm 0.02}$	$\mathbf{0.17 \pm 0.02}$
	CI-CI	0.9 ± 0.03	$\mathbf{0.87 \pm 0.03}$		CI-CI	0.17 ± 0.01	$\mathbf{0.17 \pm 0.01}$
LED ($K=10$)	RN-BGB	0.49 ± 0.01	0.49 ± 0.01	Glass ($K=6$)	RN-BGB	0.3 ± 0.02	0.3 ± 0.02
	CR-BGB	0.47 ± 0.02	0.46 ± 0.01		CR-BGB	0.2 ± 0.02	0.2 ± 0.02
	NORM-BGB	0.49 ± 0.01	0.49 ± 0.01		NORM-BGB	0.43 ± 0.06	0.41 ± 0.06
	BIN-BIN	0.49 ± 0.01	0.49 ± 0.01		BIN-BIN	0.42 ± 0.07	0.41 ± 0.07
	CI-CI	$\mathbf{0.53 \pm 0.01}$	$\mathbf{0.53 \pm 0.02}$		CI-CI	$\mathbf{0.55 \pm 0.07}$	$\mathbf{0.53 \pm 0.07}$
Breast ($K=2$)	RN-BGB	0.89 ± 0.01	$\mathbf{0.9 \pm 0.01}$	German ($K=2$)	RN-BGB	0.01 ± 0	0.01 ± 0
	CR-BGB	0.84 ± 0.01	0.85 ± 0.01		CR-BGB	0.05 ± 0.01	0.05 ± 0.01
	NORM-BGB	0.85 ± 0.01	0.85 ± 0.01		NORM-BGB	0.03 ± 0.01	0.05 ± 0.01
	BIN-BIN	$\mathbf{0.9 \pm 0.01}$	$\mathbf{0.9 \pm 0.01}$		BIN-BIN	$\mathbf{0.07 \pm 0.01}$	$\mathbf{0.07 \pm 0.01}$
	CI-CI	0.89 ± 0.05	0.89 ± 0.05		CI-CI	0.03 ± 0.03	0.04 ± 0.03
Abalone ($K=28$)	RN-BGB	$\mathbf{0.05 \pm 0}$	$\mathbf{0.05 \pm 0}$	Australian ($K=2$)	RN-BGB	0.22 ± 0.01	0.22 ± 0.01
	CR-BGB	$\mathbf{0.05 \pm 0}$	$\mathbf{0.05 \pm 0}$		CR-BGB	0.09 ± 0.01	0.09 ± 0.01
	NORM-BGB	$\mathbf{0.05 \pm 0}$	$\mathbf{0.05 \pm 0}$		NORM-BGB	$\mathbf{0.5 \pm 0.01}$	$\mathbf{0.5 \pm 0.01}$
	BIN-BIN	$\mathbf{0.05 \pm 0.01}$	$\mathbf{0.05 \pm 0.01}$		BIN-BIN	0.32 ± 0.07	0.31 ± 0.07
	CI-CI	$\mathbf{0.05 \pm 0}$	$\mathbf{0.05 \pm 0}$		CI-CI	$\mathbf{0.5 \pm 0.02}$	$\mathbf{0.5 \pm 0.02}$
Horsecolic ($K=2$)	RN-BGB	0.08 ± 0.01	0.09 ± 0.01	Mushroom ($K=2$)	RN-BGB	0.62 ± 0.01	0.62 ± 0.01
	CR-BGB	-0.04 ± 0.02	-0.06 ± 0.02		CR-BGB	0.62 ± 0.01	0.62 ± 0.01
	NORM-BGB	-0.04 ± 0.02	0.06 ± 0.02		NORM-BGB	0.62 ± 0.01	0.62 ± 0.01
	BIN-BIN	0.28 ± 0.09	0.22 ± 0.09		BIN-BIN	0.64 ± 0.01	0.64 ± 0.01
	CI-CI	$\mathbf{0.39 \pm 0.02}$	$\mathbf{0.37 \pm 0.02}$		CI-CI	$\mathbf{0.94 \pm 0}$	$\mathbf{0.94 \pm 0}$
PenDigits ($K=10$)	RN-BGB	$\mathbf{0.56 \pm 0.01}$	$\mathbf{0.56 \pm 0.01}$	Soybean ($K=19$)	RN-BGB	0.50 ± 0.03	0.50 ± 0.04
	CR-BGB	0.55 ± 0.01	0.55 ± 0.01		CR-BGB	0.50 ± 0.03	0.50 ± 0.03
	NORM-BGB	0.53 ± 0	0.53 ± 0		NORM-BGB	0.50 ± 0.03	0.50 ± 0.03
	BIN-BIN	0.44 ± 0.01	0.45 ± 0.01		BIN-BIN	0.53 ± 0.02	0.53 ± 0.02
	CI-CI	$\mathbf{0.58 \pm 0.01}$	$\mathbf{0.58 \pm 0.02}$		CI-CI	$\mathbf{0.56 \pm 0.02}$	$\mathbf{0.56 \pm 0.02}$
Segmentation ($K=7$)	RN-BGB	0.52 ± 0	0.52 ± 0.01	Tictactoe ($K=2$)	RN-BGB	0.02 ± 0.01	0.01 ± 0.01
	CR-BGB	0.46 ± 0.01	0.46 ± 0.01		CR-BGB	0.02 ± 0.01	0.01 ± 0.01
	NORM-BGB	0.5 ± 0	0.51 ± 0		NORM-BGB	0.02 ± 0.01	0.01 ± 0.01
	BIN-BIN	0.55 ± 0.01	0.54 ± 0.01		BIN-BIN	0.05 ± 0.02	0.03 ± 0.02
	CI-CI	$\mathbf{0.72 \pm 0.02}$	$\mathbf{0.70 \pm 0.02}$		CI-CI	$\mathbf{0.13 \pm 0.02}$	$\mathbf{0.11 \pm 0.02}$
Waveform ($K=3$)	RN-BGB	$\mathbf{0.26 \pm 0}$	$\mathbf{0.26 \pm 0}$	Adult ($K=2$)	RN-BGB	0.18 ± 0	0.18 ± 0.01
	CR-BGB	0.25 ± 0	0.25 ± 0		CR-BGB	0.18 ± 0	0.18 ± 0.01
	NORM-BGB	0.25 ± 0	0.25 ± 0		NORM-BGB	0.18 ± 0	0.18 ± 0.01
	BIN-BIN	0.23 ± 0.01	0.23 ± 0.01		BIN-BIN	0.18 ± 0	0.18 ± 0.01
	CI-CI	0.22 ± 0.02	0.22 ± 0.02		CI-CI	$\mathbf{0.21 \pm 0}$	$\mathbf{0.21 \pm 0.01}$
Phoneme ($K=5$)	RN-BGB	0.61 ± 0	0.61 ± 0.01				
	CR-BGB	0.64 ± 0.01	0.64 ± 0.01				
	NORM-BGB	0.65 ± 0	0.65 ± 0.01				
	BIN-BIN	0.64 ± 0	0.63 ± 0				
	CI-CI	$\mathbf{0.74 \pm 0.01}$	$\mathbf{0.7. \pm 0.01}$				

TABLE B.2 – Les performances moyennes de l'algorithme des K-moyennes précédé par différents prétraitements en utilisant l'ARI (cas où $K = J$).

Annexe C

Chapitre 4

C.1 Les performances prédictives des K-moyennes dans le Cas où K=J

Le tableau C.2 (respectivement le tableau C.3) présente les performances moyennes en termes d'ARI de l'algorithme des K-moyennes standard précédé par Conditional Info (respectivement Rank Normalization) et en utilisant les différentes méthodes d'initialisation. Les résultats présentés dans les deux tableaux (C.2 et C.3) sont obtenus lorsque l'algorithme des K-moyennes est exécuté une, dix, et cent fois.

R=1							
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**K++R**
German	0.03 ± 0.06	0.03 ± 0.06	0.04 ± 0.05	0.03 ± 0.06	0.05 ± 0.06	0.07 ± 0.04	$\mathbf{0.12 \pm 0.04}$
Australian	0.44 ± 0.11	0.44 ± 0.11	0.39 ± 0.16	0.49 ± 0.11	0.34 ± 0.19	0.35 ± 0.14	$\mathbf{0.50 \pm 0.10}$
LED	0.36 ± 0.05	0.37 ± 0.06	0.43 ± 0.04	0.41 ± 0.06	0.36 ± 0.03	0.27 ± 0.03	$\mathbf{0.51 \pm 0.05}$
Hepatitis	0.10 ± 0.11	0.10 ± 0.11	0.13 ± 0.22	0.03 ± 0.11	0.21 ± 0.26	$\mathbf{0.27 \pm 0.24}$	0.20 ± 0.17
Heart	$\mathbf{0.36 \pm 0.14}$	$\mathbf{0.36 \pm 0.15}$	0.33 ± 0.14	0.17 ± 0.15	0.34 ± 0.15	0.22 ± 0.17	$\mathbf{0.36 \pm 0.14}$
Glass	0.52 ± 0.18	0.52 ± 0.17	0.56 ± 0.17	0.79 ± 0.10	0.46 ± 0.14	0.30 ± 0.02	$\mathbf{0.82 \pm 0.09}$
Breast	0.88 ± 0.07	0.88 ± 0.07	0.87 ± 0.1	0.87 ± 0.1	0.87 ± 0.1	0.80 ± 0.23	$\mathbf{0.90 \pm 0.07}$
Iris	0.57 ± 0.15	0.57 ± 0.15	0.64 ± 0.14	0.61 ± 0.14	0.58 ± 0.10	0.60 ± 0.11	$\mathbf{0.72 \pm 0.12}$
Pima	0.03 ± 0.10	0.03 ± 0.10	0.04 ± 0.10	-0.02 ± 0.02	-0.00 ± 0.03	0.07 ± 0.13	$\mathbf{0.10 \pm 0.12}$
Wine	0.86 ± 0.10	0.85 ± 0.12	0.82 ± 0.13	0.86 ± 0.10	0.84 ± 0.14	0.66 ± 0.23	$\mathbf{0.92 \pm 0.09}$
Tictactoe	0.07 ± 0.06	0.07 ± 0.06	0.06 ± 0.06	0.11 ± 0.05	0.07 ± 0.06	0.05 ± 0.08	$\mathbf{0.14 \pm 0.04}$
Vehicle	0.18 ± 0.06	0.18 ± 0.07	0.18 ± 0.04	$\mathbf{0.19 \pm 0.04}$	0.12 ± 0.05	0.10 ± 0.07	$\mathbf{0.19 \pm 0.07}$
Horsecolic	0.30 ± 0.13	0.30 ± 0.13	0.19 ± 0.15	0.29 ± 0.16	0.14 ± 0.16	0.11 ± 0.12	$\mathbf{0.37 \pm 0.09}$
Abalone	$\mathbf{0.06 \pm 0.01}$	$\mathbf{0.06 \pm 0.01}$	0.05 ± 0.01	0.05 ± 0.01	$\mathbf{0.06 \pm 0.00}$	$\mathbf{0.06 \pm 0.01}$	$\mathbf{0.06 \pm 0.01}$
Segmentation	0.52 ± 0.06	0.52 ± 0.06	0.60 ± 0.06	0.67 ± 0.05	0.65 ± 0.04	0.63 ± 0.05	$\mathbf{0.68 \pm 0.03}$
Soybean	0.45 ± 0.07	0.45 ± 0.07	0.52 ± 0.05	0.44 ± 0.10	0.49 ± 0.09	0.42 ± 0.05	$\mathbf{0.61 \pm 0.06}$
Mushroom	0.55 ± 0.2	0.55 ± 0.2	0.69 ± 0.2	$\mathbf{0.94 \pm 0.01}$	0.08 ± 0.01	0.06 ± 0.01	$\mathbf{0.94 \pm 0.01}$
Phoneme	0.73 ± 0.01	0.73 ± 0.01	0.73 ± 0.01	0.72 ± 0.01	0.73 ± 0.01	0.73 ± 0.01	0.73 ± 0.01
PenDigits	0.51 ± 0.02	0.50 ± 0.02	0.52 ± 0.03	0.51 ± 0.01	0.53 ± 0.01	0.48 ± 0.08	$\mathbf{0.62 \pm 0.01}$
Adult	0.10 ± 0.05	0.10 ± 0.05	0.12 ± 0.06	0.15 ± 0.05	0.05 ± 0.06	0.04 ± 0.06	$\mathbf{0.17 \pm 0.02}$
Waveform	0.19 ± 0.03	0.19 ± 0.03	0.22 ± 0.01	0.22 ± 0.01	0.21 ± 0.01	0.14 ± 0.05	$\mathbf{0.23 \pm 0.01}$

TABLE C.1 – Les performances prédictives de l'algorithme des K-moyennes précédé par CI dans le cas où $K = J$ pour $R = 1$

R=10							
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**K++R**
German	0.04 ± 0.06	0.04 ± 0.06	0.04 ± 0.06	0.03 ± 0.06	0.05 ± 0.05	0.07 ± 0.04	$\mathbf{0.12 \pm 0.04}$
Australian	$\mathbf{0.5 \pm 0.10}$	$\mathbf{0.5 \pm 0.10}$	$\mathbf{0.5 \pm 0.10}$	0.49 ± 0.11	0.49 ± 0.10	0.35 ± 0.14	$\mathbf{0.5 \pm 0.10}$
LED	0.43 ± 0.03	0.43 ± 0.03	0.45 ± 0.04	0.41 ± 0.06	0.43 ± 0.05	0.27 ± 0.03	$\mathbf{0.53 \pm 0.05}$
Hepatitis	0.07 ± 0.12	0.05 ± 0.10	0.05 ± 0.10	0.03 ± 0.11	0.11 ± 0.17	$\mathbf{0.27 \pm 0.24}$	0.20 ± 0.17
Heart	0.31 ± 0.18	0.31 ± 0.18	0.31 ± 0.18	0.17 ± 0.15	0.31 ± 0.18	0.22 ± 0.17	$\mathbf{0.36 \pm 0.14}$
Glass	0.47 ± 0.11	0.51 ± 0.07	0.54 ± 0.16	0.79 ± 0.10	0.52 ± 0.16	0.30 ± 0.02	$\mathbf{0.82 \pm 0.09}$
Breast	$\mathbf{0.88 \pm 0.07}$	$\mathbf{0.88 \pm 0.07}$	0.87 ± 0.10	0.87 ± 0.10	0.87 ± 0.10	0.8 ± 0.23	$\mathbf{0.88 \pm 0.07}$
Iris	0.62 ± 0.13	0.62 ± 0.13	0.62 ± 0.13	0.62 ± 0.14	0.63 ± 0.12	0.60 ± 0.11	$\mathbf{0.72 \pm 0.12}$
Pima	-0.03 ± 0.02	-0.03 ± 0.02	-0.0 ± 0.03	-0.02 ± 0.02	-0.02 ± 0.03	0.07 ± 0.13	$\mathbf{0.10 \pm 0.12}$
Wine	0.86 ± 0.10	0.86 ± 0.10	0.91 ± 0.09	0.86 ± 0.10	0.86 ± 0.10	0.66 ± 0.23	$\mathbf{0.92 \pm 0.09}$
Tictactoe	0.11 ± 0.05	0.11 ± 0.05	0.10 ± 0.04	0.11 ± 0.05	0.09 ± 0.05	0.05 ± 0.08	$\mathbf{0.14 \pm 0.04}$
Vehicle	0.16 ± 0.04	0.16 ± 0.04	0.17 ± 0.04	$\mathbf{0.19 \pm 0.04}$	0.13 ± 0.03	0.1 ± 0.07	$\mathbf{0.19 \pm 0.07}$
Horsecolic	$\mathbf{0.37 \pm 0.09}$	$\mathbf{0.37 \pm 0.09}$	$\mathbf{0.37 \pm 0.09}$	0.29 ± 0.16	0.14 ± 0.16	0.11 ± 0.12	$\mathbf{0.37 \pm 0.09}$
Abalone	0.05 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	$\mathbf{0.06 \pm 0.01}$	$\mathbf{0.06 \pm 0.01}$
Segmentation	$\mathbf{0.68 \pm 0.05}$	0.66 ± 0.05	0.70 ± 0.05	0.67 ± 0.05	0.71 ± 0.04	0.63 ± 0.05	$\mathbf{0.68 \pm 0.03}$
Soybean	0.48 ± 0.08	0.49 ± 0.08	0.52 ± 0.11	0.44 ± 0.06	0.50 ± 0.08	0.42 ± 0.05	$\mathbf{0.62 \pm 0.10}$
Mushroom	0.94 ± 0.01	0.94 ± 0.01	0.94 ± 0.01	0.94 ± 0.01	$\mathbf{0.97 \pm 0.01}$	0.04 ± 0.01	0.94 ± 0.01
Phoneme	$\mathbf{0.73 \pm 0.01}$	$\mathbf{0.73 \pm 0.01}$	$\mathbf{0.73 \pm 0.02}$	0.72 ± 0.01	$\mathbf{0.73 \pm 0.01}$	$\mathbf{0.73 \pm 0.01}$	$\mathbf{0.73 \pm 0.01}$
PenDigits	0.58 ± 0.02	0.58 ± 0.02	0.57 ± 0.02	0.51 ± 0.02	0.55 ± 0.01	0.48 ± 0.05	$\mathbf{0.62 \pm 0.01}$
Adult	0.15 ± 0.01	0.15 ± 0.01	0.13 ± 0.01	0.15 ± 0.01	0.14 ± 0.01	0.04 ± 0.01	$\mathbf{0.17 \pm 0.01}$
Waveform	0.22 ± 0.02	0.22 ± 0.02	0.22 ± 0.02	0.22 ± 0.03	0.22 ± 0.03	0.14 ± 0.09	$\mathbf{0.23 \pm 0.03}$

R=100							
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**K++R**
German	0.04 ± 0.06	0.04 ± 0.06	0.04 ± 0.06	0.03 ± 0.06	0.05 ± 0.06	0.07 ± 0.04	$\mathbf{0.12 \pm 0.04}$
Australian	$\mathbf{0.5 \pm 0.10}$	$\mathbf{0.5 \pm 0.10}$	$\mathbf{0.5 \pm 0.10}$	0.49 ± 0.11	$\mathbf{0.5 \pm 0.10}$	0.35 ± 0.14	$\mathbf{0.5 \pm 0.10}$
LED	0.47 ± 0.06	0.48 ± 0.05	0.48 ± 0.05	0.41 ± 0.06	0.44 ± 0.04	0.27 ± 0.03	$\mathbf{0.53 \pm 0.05}$
Hepatitis	0.05 ± 0.10	0.0 ± 0.10	0.05 ± 0.10	0.03 ± 0.11	0.08 ± 0.16	$\mathbf{0.27 \pm 0.24}$	0.20 ± 0.17
Heart	0.30 ± 0.17	0.30 ± 0.17	0.30 ± 0.17	0.17 ± 0.15	0.30 ± 0.17	0.22 ± 0.17	$\mathbf{0.36 \pm 0.14}$
Glass	0.52 ± 0.19	0.52 ± 0.19	0.53 ± 0.17	0.79 ± 0.10	0.49 ± 0.15	0.30 ± 0.02	$\mathbf{0.82 \pm 0.09}$
Breast	0.87 ± 0.10	0.87 ± 0.10	0.87 ± 0.10	0.87 ± 0.10	0.87 ± 0.10	0.8 ± 0.23	$\mathbf{0.88 \pm 0.07}$
Iris	0.62 ± 0.13	0.62 ± 0.13	0.62 ± 0.13	0.61 ± 0.14	0.62 ± 0.13	0.60 ± 0.11	$\mathbf{0.72 \pm 0.12}$
Pima	-0.02 ± 0.03	-0.02 ± 0.03	-0.02 ± 0.03	-0.02 ± 0.02	-0.02 ± 0.03	0.07 ± 0.13	$\mathbf{0.10 \pm 0.12}$
Wine	0.91 ± 0.09	0.91 ± 0.09	0.91 ± 0.09	0.86 ± 0.10	0.86 ± 0.10	0.66 ± 0.23	$\mathbf{0.92 \pm 0.09}$
Tictactoe	0.11 ± 0.05	0.11 ± 0.05	0.11 ± 0.05	0.11 ± 0.05	0.10 ± 0.04	0.05 ± 0.08	$\mathbf{0.14 \pm 0.04}$
Vehicle	0.17 ± 0.03	0.17 ± 0.03	0.17 ± 0.03	$\mathbf{0.19 \pm 0.04}$	0.14 ± 0.03	0.1 ± 0.07	$\mathbf{0.19 \pm 0.07}$
Horsecolic	$\mathbf{0.37 \pm 0.09}$	$\mathbf{0.37 \pm 0.09}$	$\mathbf{0.37 \pm 0.09}$	0.29 ± 0.16	0.14 ± 0.16	0.11 ± 0.12	$\mathbf{0.37 \pm 0.09}$
Abalone	0.05 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	$\mathbf{0.06 \pm 0.01}$	$\mathbf{0.06 \pm 0.01}$	$\mathbf{0.06 \pm 0.01}$	$\mathbf{0.06 \pm 0.01}$
Segmentation	$\mathbf{0.70 \pm 0.06}$	$\mathbf{0.70 \pm 0.06}$	$\mathbf{0.70 \pm 0.06}$	0.67 ± 0.05	$\mathbf{0.7 \pm 0.05}$	0.63 ± 0.05	0.68 ± 0.3
Soybean	0.52 ± 0.07	0.53 ± 0.08	0.53 ± 0.10	0.45 ± 0.04	0.51 ± 0.07	0.42 ± 0.05	$\mathbf{0.61 \pm 0.06}$
Adult	0.15 ± 0.01	0.15 ± 0.01	0.15 ± 0.01	0.15 ± 0.01	0.15 ± 0.01	0.04 ± 0.01	0.17 ± 0.01
Phoneme	0.73 ± 0.01	0.73 ± 0.01	0.73 ± 0.02	0.72 ± 0.01	0.73 ± 0.01	0.73 ± 0.01	0.73 ± 0.01
PenDigits	0.56 ± 0.02	0.56 ± 0.02	0.56 ± 0.02	0.51 ± 0.02	0.56 ± 0.01	0.48 ± 0.05	$\mathbf{0.62 \pm 0.01}$
Mushroom	0.94 ± 0.01	0.94 ± 0.01	0.94 ± 0.01	0.94 ± 0.01	0.94 ± 0.01	0.06 ± 0.01	0.94 ± 0.01
Waveform	0.22 ± 0.02	0.22 ± 0.02	0.22 ± 0.02	0.22 ± 0.02	0.22 ± 0.02	0.14 ± 0.06	$\mathbf{0.23 \pm 0.02}$

TABLE C.2 – Les performances prédictives de l'algorithme des K-moyennes précédé par CI dans le cas où $K = J$ pour $R \in \{10, 100\}$

R=1							
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**K++R**
German	0.02 ± 0.02	0.02 ± 0.02	$\mathbf{0.03 \pm 0.04}$	0.01 ± 0.02	0.01 ± 0.01	0.02 ± 0.03	0.01 ± 0.01
Australian	0.22 ± 0.04	$\mathbf{0.22 \pm 0.04}$	0.22 ± 0.04	$\mathbf{0.22 \pm 0.04}$	0.20 ± 0.08	0.04 ± 0.09	$\mathbf{0.22 \pm 0.04}$
LED	0.36 ± 0.05	0.37 ± 0.05	0.43 ± 0.06	0.45 ± 0.03	0.38 ± 0.06	0.25 ± 0.06	$\mathbf{0.51 \pm 0.04}$
Hepatitis	0.17 ± 0.20	0.17 ± 0.20	$\mathbf{0.19 \pm 0.20}$	0.17 ± 0.17	0.15 ± 0.22	-0.01 ± 0.05	$\mathbf{0.19 \pm 0.19}$
Heart	$\mathbf{0.40 \pm 0.10}$	$\mathbf{0.40 \pm 0.10}$	$\mathbf{0.40 \pm 0.09}$	0.35 ± 0.08	$\mathbf{0.40 \pm 0.09}$	0.25 ± 0.22	$\mathbf{0.40 \pm 0.10}$
Glass	0.2 ± 0.08	0.21 ± 0.07	0.26 ± 0.09	0.27 ± 0.08	0.20 ± 0.08	0.23 ± 0.08	$\mathbf{0.29 \pm 0.11}$
Breast	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$
Iris	$\mathbf{0.65 \pm 0.13}$	$\mathbf{0.65 \pm 0.13}$	0.64 ± 0.15	0.64 ± 0.13	$\mathbf{0.65 \pm 0.14}$	$\mathbf{0.65 \pm 0.15}$	0.64 ± 0.14
Pima	0.10 ± 0.04	0.11 ± 0.05	0.07 ± 0.04	0.09 ± 0.04	0.1 ± 0.04	0.10 ± 0.05	$\mathbf{0.11 \pm 0.04}$
Wine	0.86 ± 0.10	0.86 ± 0.10	$\mathbf{0.88 \pm 0.10}$	0.87 ± 0.08	0.87 ± 0.08	0.87 ± 0.07	0.87 ± 0.09
Tictactoe	0.01 ± 0.02	0.02 ± 0.02	0.02 ± 0.02	$\mathbf{0.07 \pm 0.02}$	0.02 ± 0.02	0.01 ± 0.02	$\mathbf{0.07 \pm 0.04}$
Vehicle	0.1 ± 0.02	0.10 ± 0.02	0.09 ± 0.01	0.08 ± 0.01	0.08 ± 0.01	0.08 ± 0.0	$\mathbf{0.12 \pm 0.02}$
Horsecolic	0.08 ± 0.08	0.08 ± 0.08	0.09 ± 0.07	0.08 ± 0.07	0.11 ± 0.07	0.02 ± 0.03	$\mathbf{0.12 \pm 0.06}$
Abalone	0.04 ± 0.00	0.04 ± 0.00	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.05 \pm 0.01}$	$\mathbf{0.05 \pm 0.01}$	$\mathbf{0.05 \pm 0.01}$	$\mathbf{0.05 \pm 0.00}$
Segmentation	0.5 ± 0.03	0.5 ± 0.03	0.50 ± 0.04	0.50 ± 0.02	0.50 ± 0.02	0.52 ± 0.01	$\mathbf{0.54 \pm 0.02}$
Soybean	0.44 ± 0.04	0.44 ± 0.04	0.46 ± 0.08	0.41 ± 0.08	0.52 ± 0.05	0.38 ± 0.03	$\mathbf{0.64 \pm 0.06}$
Adult	0.14 ± 0.08	0.14 ± 0.08	0.14 ± 0.08	0.00 ± 0.00	0.14 ± 0.08	0.16 ± 0.07	$\mathbf{0.18 \pm 0.01}$
Phoneme	$\mathbf{0.61 \pm 0.02}$	$\mathbf{0.61 \pm 0.02}$	0.59 ± 0.07	$\mathbf{0.61 \pm 0.02}$	$\mathbf{0.61 \pm 0.02}$	0.46 ± 0.11	$\mathbf{0.61 \pm 0.02}$
PenDigits	0.55 ± 0.05	0.55 ± 0.05	0.56 ± 0.06	0.59 ± 0.04	0.57 ± 0.03	0.57 ± 0.04	$\mathbf{0.63 \pm 0.01}$
Mushroom	0.20 ± 0.17	0.20 ± 0.17	0.52 ± 0.20	0.62 ± 0.02	0.39 ± 0.23	0.00 ± 0.00	$\mathbf{0.62 \pm 0.02}$
Waveform	0.26 ± 0.01	0.26 ± 0.01	0.26 ± 0.01	0.26 ± 0.01	0.26 ± 0.01	0.26 ± 0.01	$\mathbf{0.28 \pm 0.01}$

R=10							
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**K++R**
German	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.02	0.01 ± 0.01	$\mathbf{0.02 \pm 0.03}$	0.01 ± 0.01
Australian	$\mathbf{0.22 \pm 0.04}$	$\mathbf{0.22 \pm 0.04}$	$\mathbf{0.22 \pm 0.04}$	$\mathbf{0.22 \pm 0.04}$	$\mathbf{0.22 \pm 0.04}$	0.04 ± 0.09	$\mathbf{0.22 \pm 0.04}$
LED	0.46 ± 0.05	0.45 ± 0.05	0.47 ± 0.05	0.45 ± 0.03	0.43 ± 0.04	0.32 ± 0.05	$\mathbf{0.51 \pm 0.04}$
Hepatitis	0.17 ± 0.17	0.17 ± 0.17	0.17 ± 0.17	0.17 ± 0.17	0.17 ± 0.17	-0.01 ± 0.05	$\mathbf{0.19 \pm 0.19}$
Heart	$\mathbf{0.40 \pm 0.10}$	$\mathbf{0.40 \pm 0.10}$	$\mathbf{0.40 \pm 0.10}$	0.39 ± 0.08	$\mathbf{0.40 \pm 0.10}$	0.25 ± 0.22	$\mathbf{0.40 \pm 0.10}$
Glass	$\mathbf{0.3 \pm 0.09}$	$\mathbf{0.3 \pm 0.09}$	0.26 ± 0.09	0.25 ± 0.06	0.24 ± 0.08	0.23 ± 0.06	$\mathbf{0.3 \pm 0.11}$
Breast	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$
Iris	0.66 ± 0.15	0.66 ± 0.15	$\mathbf{0.67 \pm 0.16}$	0.64 ± 0.13	0.66 ± 0.15	0.65 ± 0.15	0.64 ± 0.14
Pima	0.10 ± 0.05	0.10 ± 0.05	0.09 ± 0.04	0.09 ± 0.04	0.09 ± 0.05	0.10 ± 0.05	$\mathbf{0.11 \pm 0.04}$
Wine	0.88 ± 0.10	0.88 ± 0.10	$\mathbf{0.90 \pm 0.08}$	0.87 ± 0.08	0.88 ± 0.10	$\mathbf{0.9 \pm 0.07}$	$\mathbf{0.9 \pm 0.09}$
Tictactoe	0.01 ± 0.02	0.01 ± 0.02	0.02 ± 0.02	$\mathbf{0.07 \pm 0.02}$	0.01 ± 0.02	0.01 ± 0.02	$\mathbf{0.07 \pm 0.04}$
Vehicle	0.09 ± 0.01	0.08 ± 0.01	0.08 ± 0.02	0.08 ± 0.02	0.08 ± 0.01	0.08 ± 0.02	$\mathbf{0.12 \pm 0.02}$
Horsecolic	0.10 ± 0.10	0.09 ± 0.07	0.09 ± 0.07	0.08 ± 0.07	0.08 ± 0.07	0.02 ± 0.03	$\mathbf{0.12 \pm 0.06}$
Abalone	$\mathbf{0.05 \pm 0.01}$	$\mathbf{0.05 \pm 0.01}$	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.05 \pm 0.00}$
Segmentation	0.52 ± 0.01	0.52 ± 0.01	0.53 ± 0.01	0.50 ± 0.02	0.52 ± 0.02	0.52 ± 0.01	$\mathbf{0.54 \pm 0.02}$
Soybean	0.5 ± 0.05	0.52 ± 0.06	0.50 ± 0.06	0.39 ± 0.05	0.51 ± 0.07	0.39 ± 0.03	$\mathbf{0.64 \pm 0.06}$
Adult	0.18 ± 0.01	0.18 ± 0.01	0.18 ± 0.01	0.00 ± 0.00	0.18 ± 0.01	0.16 ± 0.03	0.18 ± 0.01
Phoneme	$\mathbf{0.61 \pm 0.02}$	$\mathbf{0.61 \pm 0.02}$	$\mathbf{0.61 \pm 0.02}$	$\mathbf{0.61 \pm 0.02}$	$\mathbf{0.61 \pm 0.02}$	0.46 ± 0.11	$\mathbf{0.61 \pm 0.02}$
PenDigits	0.57 ± 0.02	0.57 ± 0.02	0.57 ± 0.02	0.59 ± 0.02	0.56 ± 0.02	0.57 ± 0.04	$\mathbf{0.63 \pm 0.02}$
Mushroom	$\mathbf{0.62 \pm 0.02}$	$\mathbf{0.62 \pm 0.02}$	$\mathbf{0.62 \pm 0.02}$	$\mathbf{0.62 \pm 0.02}$	$\mathbf{0.62 \pm 0.02}$	0.00 ± 0.00	$\mathbf{0.62 \pm 0.02}$
Waveform	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	$\mathbf{0.28 \pm 0.00}$

R=100							
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**K++R**
German	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.02	0.01 ± 0.01	$\mathbf{0.02 \pm 0.03}$	0.01 ± 0.01
Australian	0.22 ± 0.04	$\mathbf{0.22 \pm 0.04}$	$\mathbf{0.22 \pm 0.04}$	$\mathbf{0.22 \pm 0.04}$	$\mathbf{0.22 \pm 0.04}$	0.04 ± 0.09	$\mathbf{0.22 \pm 0.04}$
LED	0.47 ± 0.03	0.47 ± 0.03	0.48 ± 0.03	0.45 ± 0.02	0.47 ± 0.02	0.36 ± 0.03	$\mathbf{0.51 \pm 0.01}$
Hepatitis	0.17 ± 0.17	0.17 ± 0.17	0.17 ± 0.17	0.17 ± 0.17	0.17 ± 0.17	-0.01 ± 0.05	$\mathbf{0.19 \pm 0.19}$
Heart	$\mathbf{0.40 \pm 0.10}$	$\mathbf{0.40 \pm 0.10}$	$\mathbf{0.40 \pm 0.10}$	$\mathbf{0.39 \pm 0.08}$	$\mathbf{0.40 \pm 0.10}$	0.25 ± 0.22	$\mathbf{0.40 \pm 0.10}$
Glass	0.29 ± 0.09	0.29 ± 0.08	$\mathbf{0.3 \pm 0.07}$	0.26 ± 0.08	0.24 ± 0.06	0.23 ± 0.06	$\mathbf{0.3 \pm 0.11}$
Breast	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.90 \pm 0.05}$
Iris	$\mathbf{0.66 \pm 0.15}$	$\mathbf{0.66 \pm 0.15}$	$\mathbf{0.66 \pm 0.15}$	0.64 ± 0.13	$\mathbf{0.66 \pm 0.15}$	0.65 ± 0.15	0.64 ± 0.14
Pima	0.08 ± 0.04	0.08 ± 0.04	0.08 ± 0.04	0.09 ± 0.04	0.08 ± 0.04	0.10 ± 0.05	$\mathbf{0.11 \pm 0.04}$
Wine	$\mathbf{0.90 \pm 0.08}$	$\mathbf{0.90 \pm 0.08}$	$\mathbf{0.90 \pm 0.08}$	0.87 ± 0.08	0.88 ± 0.10	0.87 ± 0.07	$\mathbf{0.9 \pm 0.09}$
Tictactoe	0.01 ± 0.02	0.01 ± 0.02	0.01 ± 0.02	$\mathbf{0.07 \pm 0.02}$	0.01 ± 0.02	0.01 ± 0.02	$\mathbf{0.07 \pm 0.04}$
Vehicle	0.08 ± 0.01	0.08 ± 0.02	0.08 ± 0.01	0.08 ± 0.02	0.08 ± 0.01	0.08 ± 0.02	$\mathbf{0.12 \pm 0.02}$
Horsecolic	0.08 ± 0.07	0.09 ± 0.08	0.09 ± 0.07	0.08 ± 0.07	0.08 ± 0.07	0.02 ± 0.03	$\mathbf{0.12 \pm 0.08}$
Abalone	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.05 \pm 0.01}$	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.05 \pm 0.01}$
Segmentation	0.52 ± 0.01	0.52 ± 0.01	0.52 ± 0.01	0.50 ± 0.02	0.52 ± 0.01	0.52 ± 0.01	$\mathbf{0.54 \pm 0.02}$
Soybean	0.52 ± 0.07	0.51 ± 0.08	0.5 ± 0.07	0.38 ± 0.05	0.52 ± 0.05	0.39 ± 0.03	$\mathbf{0.64 \pm 0.06}$
Adult	$\mathbf{0.18 \pm 0.01}$	$\mathbf{0.18 \pm 0.01}$	$\mathbf{0.18 \pm 0.01}$	0.00 ± 0.00	$\mathbf{0.18 \pm 0.01}$	0.16 ± 0.03	$\mathbf{0.18 \pm 0.01}$
Phoneme	$\mathbf{0.61 \pm 0.02}$	$\mathbf{0.61 \pm 0.02}$	$\mathbf{0.61 \pm 0.02}$	$\mathbf{0.61 \pm 0.02}$	$\mathbf{0.61 \pm 0.02}$	0.46 ± 0.11	$\mathbf{0.61 \pm 0.02}$
PenDigits	0.57 ± 0.02	0.57 ± 0.02	0.57 ± 0.02	0.59 ± 0.02	0.56 ± 0.02	0.57 ± 0.04	$\mathbf{0.63 \pm 0.02}$
Mushroom	$\mathbf{0.62 \pm 0.02}$	$\mathbf{0.62 \pm 0.02}$	$\mathbf{0.62 \pm 0.02}$	$\mathbf{0.62 \pm 0.02}$	$\mathbf{0.62 \pm 0.02}$	0.00 ± 0.00	$\mathbf{0.62 \pm 0.02}$
Waveform	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	$\mathbf{0.28 \pm 0.00}$

TABLE C.3 – Les performances prédictives de l'algorithme des K-moyennes précédé par RN dans le cas où $K = J$

C.2 Les aires sous les courbes d'ARI (ALC-ARI)

Le tableau C.4 (respectivement le tableau C.5) présente les performances moyennes en termes d'ALC-ARI de l'algorithme des K-moyennes standard précédé par Conditional Info (respectivement Rank Normalization) et en utilisant les différentes méthodes d'initialisation. Les résultats présentés dans les deux tableaux (C.4 et C.5) sont obtenus lorsque l'algorithme des K-moyennes est exécuté une, dix, et cent fois.

R=1										
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**CD**	**K++R**	**SB**	**RS**
Iris	0.56	0.57	0.64	0.62	0.58	0.58	**0.72**	0.69	0.71	0.71
Hepatitis	0.08	0.08	0.09	0.10	0.12	**0.16**	0.11	0.10	0.12	0.12
Wine	0.39	0.39	0.40	0.46	0.49	0.47	0.37	0.43	0.55	**0.57**
Glass	0.43	0.43	0.42	0.61	0.46	0.36	0.57	0.50	0.61	**0.63**
Heart	0.10	0.11	0.10	0.11	0.11	0.11	0.10	0.11	0.11	**0.12**
Horsecolic	0.25	0.25	0.22	0.24	0.21	0.18	**0.31**	0.28	0.28	0.27
Soybean	0.46	0.46	0.51	0.45	0.51	0.43	**0.61**	**0.61**	**0.61**	**0.61**
Breast	0.36	0.43	0.48	0.55	0.56	**0.59**	0.44	0.46	0.58	0.58
Australian	0.09	0.10	0.09	**0.12**	0.11	0.11	0.10	0.09	0.11	0.11
Pima	**0.06**	**0.06**	0.05	0.03	0.04	0.04	**0.06**	0.05	**0.06**	**0.06**
Vehicle	0.17	0.17	0.17	0.19	0.17	0.14	0.18	0.18	0.19	**0.20**
Tictactoe	0.04	0.04	0.04	0.04	0.04	0.04	**0.05**	**0.05**	**0.05**	**0.05**
LED	0.42	0.43	0.43	0.45	0.38	0.33	**0.48**	0.47	0.47	**0.48**
German	0.03	0.03	0.03	0.03	0.04	0.05	0.06	0.05	0.06	**0.07**
Segmentation	0.39	0.40	0.42	0.45	0.63	0.51	**0.65**	0.49	0.47	0.40
Abalone	0.05	0.05	0.04	0.05	0.05	0.05	0.05	0.05	0.05	**0.06**
Mushroom	0.53	0.53	0.53	**0.64**	0.38	0.09	0.57	0.61	0.62	0.60
PenDigits	0.39	0.39	0.39	0.41	0.41	0.43	0.42	0.40	**0.45**	**0.45**
Waveform	0.09	0.09	0.09	0.09	**0.10**	**0.10**	**0.10**	0.09	**0.10**	**0.10**

R=10										
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**CD**	**K++R**	**SB**	**RS**
Iris	0.64	0.63	0.62	0.62	0.61	0.58	**0.72**	0.69	0.71	0.71
Hepatitis	0.08	0.08	0.08	0.10	0.10	**0.16**	0.11	0.09	0.12	0.12
Wine	0.36	0.37	0.38	0.47	0.46	0.47	0.37	0.39	0.55	**0.57**
Glass	0.41	0.41	0.41	0.61	0.45	0.36	0.57	0.46	0.61	**0.63**
Heart	0.10	0.10	0.09	0.11	0.09	0.11	0.10	0.10	0.11	**0.12**
Horsecolic	0.24	0.24	0.25	0.24	0.21	0.18	**0.31**	0.26	0.28	0.27
Soybean	0.50	0.51	0.53	0.45	0.52	0.43	**0.61**	0.60	**0.61**	**0.61**
Breast	0.38	0.41	0.45	0.55	0.51	**0.59**	0.43	0.45	0.57	0.58
Australian	0.09	0.09	0.09	**0.12**	0.11	0.11	0.09	0.09	0.11	0.11
Pima	0.05	0.05	0.04	0.03	0.04	0.04	**0.06**	0.05	**0.06**	**0.06**
Vehicle	0.17	0.17	0.18	0.19	0.19	0.14	0.18	0.18	0.19	**0.20**
Tictactoe	0.04	0.04	0.04	0.04	0.04	0.04	**0.05**	0.04	**0.05**	**0.05**
LED	0.44	0.45	0.44	0.45	0.41	0.33	**0.48**	0.47	0.47	**0.48**
German	0.04	0.03	0.03	0.03	0.04	0.05	0.06	0.05	0.06	**0.07**
Segmentation	0.40	0.40	0.41	0.43	0.62	0.51	**0.65**	0.47	0.47	0.40
Abalone	0.05	0.05	0.04	0.05	0.05	0.05	0.05	0.05	0.05	**0.06**
Mushroom	0.54	0.54	0.53	**0.64**	0.56	0.09	0.56	0.57	0.61	0.60
PenDigits	0.38	0.38	0.39	0.41	0.40	0.43	0.42	0.40	**0.45**	**0.45**
Waveform	0.09	0.09	0.09	0.09	**0.10**	**0.10**	**0.10**	0.09	**0.10**	**0.10**

R=100										
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**CD**	**K++R**	**SB**	**RS**
Iris	0.62	0.62	0.62	0.62	0.60	0.58	**0.72**	0.69	0.71	0.71
Hepatitis	0.08	0.08	0.08	0.10	0.10	0.16	0.11	0.09	**0.12**	**0.12**
Wine	0.36	0.36	0.37	0.46	0.45	0.47	0.37	0.38	0.55	**0.57**
Glass	0.41	0.41	0.41	0.61	0.44	0.36	0.57	0.44	0.61	**0.63**
Heart	0.09	0.10	0.09	0.11	0.09	0.11	0.10	0.09	0.11	**0.12**
Horsecolic	0.24	0.24	0.24	0.25	0.20	0.18	0.31	0.26	**0.28**	0.27
Soybean	0.52	0.51	0.54	0.45	0.52	0.43	**0.61**	0.60	**0.61**	**0.61**
Breast	0.39	0.41	0.43	0.55	0.47	0.59	0.43	0.41	0.56	**0.58**
Australian	0.09	0.09	0.09	**0.12**	0.10	0.11	0.09	0.09	0.11	0.11
Pima	0.04	0.04	0.04	0.03	0.04	0.04	**0.06**	0.05	**0.06**	**0.06**
Vehicle	0.17	0.17	0.18	0.19	**0.20**	0.14	0.18	0.18	0.19	**0.20**
Tictactoe	0.04	0.04	0.04	0.04	0.04	0.04	**0.05**	0.04	**0.05**	**0.05**
LED	0.45	0.45	0.45	0.45	0.43	0.33	**0.48**	0.46	0.46	**0.48**
German	0.03	0.03	0.03	0.03	0.04	0.05	0.06	0.05	0.06	**0.07**
Segmentation	0.40	0.40	0.41	0.42	0.62	0.51	**0.65**	0.46	0.47	0.40
Abalone	0.05	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.05	**0.06**
Mushroom	0.52	0.52	0.52	**0.64**	0.61	0.09	0.53	0.61	0.56	0.60
PenDigits	0.38	0.38	0.39	0.41	0.40	0.43	0.39	**0.45**	0.42	**0.45**
Waveform	0.09	0.09	0.09	0.09	0.09	0.10	0.09	0.10	0.10	**0.11**

TABLE C.4 – Les valeurs de l'ALC-ARI pour CI

R=1										
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**CD**	**K++R**	**SB**	**RS**
Iris	0.40	0.40	0.41	0.43	0.41	0.42	0.44	0.41	**0.45**	**0.45**
Hepatitis	0.06	0.06	0.06	0.05	0.07	**0.10**	0.09	0.06	0.09	**0.10**
Wine	0.20	0.20	0.22	0.25	0.24	0.23	0.22	0.24	0.26	**0.28**
Glass	0.20	0.20	0.21	0.23	0.23	0.21	**0.30**	0.22	**0.30**	0.27
Heart	0.06	0.07	0.07	0.07	0.07	**0.08**	0.07	0.07	**0.08**	**0.08**
Horsecolic	0.05	0.05	0.06	0.05	0.07	0.05	**0.10**	0.09	**0.10**	0.08
Soybean	0.35	0.35	0.39	0.41	0.53	0.49	**0.60**	0.46	0.57	**0.60**
Breast	0.13	0.13	0.15	0.18	0.19	0.20	0.15	0.15	0.22	**0.24**
Australian	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.06	**0.08**	0.07
Pima	0.02	0.02	0.02	**0.03**	**0.03**	**0.03**	**0.03**	0.02	**0.03**	**0.03**
Vehicle	0.09	0.09	0.09	0.08	0.09	0.09	0.10	0.09	**0.11**	0.10
Tictactoe	0.03	0.03	0.04	0.03	0.04	0.04	**0.06**	0.04	0.05	0.04
LED	0.41	0.42	0.43	0.47	0.37	0.33	**0.49**	0.47	0.47	0.48
German	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
Segmentation	0.38	0.38	0.39	0.40	0.40	0.39	0.40	0.40	0.48	**0.49**
Abalone	0.04	0.04	0.04	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	0.04	0.04
Mushroom	0.10	0.10	0.11	0.16	0.15	0.13	0.13	0.12	0.14	**0.24**
PenDigits	0.41	0.41	0.41	0.42	0.43	0.43	0.44	0.42	**0.47**	0.46
Waveform	0.10	0.10	0.10	0.10	0.10	0.10	**0.11**	0.10	0.10	0.10

R=10										
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**CD**	**K++R**	**SB**	**RS**
Iris	0.40	0.41	0.40	0.43	0.41	0.42	0.44	0.41	**0.45**	**0.45**
Hepatitis	0.05	0.05	0.06	0.05	0.07	**0.10**	0.08	0.06	0.09	**0.10**
Wine	0.21	0.21	0.21	0.25	0.23	0.23	0.21	0.23	0.25	**0.28**
Glass	0.21	0.21	0.21	0.23	0.23	0.21	**0.30**	0.22	**0.30**	0.27
Heart	0.07	0.07	0.07	0.07	0.07	**0.08**	0.07	0.07	**0.08**	**0.08**
Horsecolic	0.06	0.06	0.06	0.05	0.06	0.05	**0.10**	0.08	0.09	0.08
Soybean	0.36	0.36	0.39	0.40	0.51	0.49	**0.60**	0.44	0.57	**0.60**
Breast	0.13	0.13	0.14	0.18	0.17	0.20	0.15	0.14	0.21	**0.24**
Australian	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.06	**0.08**	0.07
Pima	0.02	0.02	0.02	**0.03**	**0.03**	**0.03**	**0.03**	0.02	**0.03**	**0.03**
Vehicle	0.09	0.09	0.09	0.08	0.09	0.09	**0.10**	0.09	**0.10**	**0.10**
Tictactoe	0.04	0.04	0.04	0.03	0.04	0.04	**0.05**	0.04	**0.05**	0.04
LED	0.46	0.46	0.45	0.47	0.40	0.36	**0.48**	**0.48**	0.47	**0.48**
German	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
Segmentation	0.39	0.39	0.39	0.40	0.40	0.39	0.40	0.40	0.47	**0.49**
Abalone	0.04	0.04	0.04	**0.05**	**0.05**	**0.05**	**0.05**	0.04	0.04	0.04
Mushroom	0.12	0.12	0.12	0.16	0.15	0.13	0.13	0.12	0.14	**0.24**
PenDigits	0.41	0.41	0.41	0.42	0.42	0.43	0.44	0.41	**0.47**	0.46
Waveform	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**

R=100										
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**CD**	**K++R**	**SB**	**RS**
Iris	0.41	0.40	0.40	0.43	0.41	0.42	0.44	0.40	**0.45**	**0.45**
Hepatitis	0.05	0.05	0.06	0.05	0.06	**0.10**	0.08	0.06	0.08	**0.10**
Wine	0.21	0.21	0.21	0.26	0.23	0.23	0.22	0.22	0.25	**0.28**
Glass	0.21	0.21	0.22	0.23	0.23	0.21	**0.30**	0.22	**0.30**	0.27
Heart	0.07	0.07	0.07	0.07	0.07	**0.08**	0.07	0.07	0.07	**0.08**
Horsecolic	0.06	0.06	0.06	0.05	0.06	0.05	**0.10**	0.08	0.09	0.08
Soybean	0.38	0.37	0.40	0.40	0.51	0.48	**0.60**	0.44	0.56	**0.60**
Breast	0.13	0.13	0.14	0.18	0.16	0.20	0.15	0.14	0.20	**0.24**
Australian	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.06	**0.08**	0.07
Pima	0.02	0.02	0.02	**0.03**	**0.03**	**0.03**	**0.03**	0.02	**0.03**	**0.03**
Vehicle	0.09	0.09	0.09	0.08	0.09	0.09	**0.10**	0.09	**0.10**	**0.10**
Tictactoe	0.04	0.04	0.04	0.03	0.04	0.04	**0.05**	**0.05**	**0.05**	0.04
LED	0.46	0.46	0.45	0.47	0.43	0.37	**0.48**	**0.48**	0.47	**0.48**
German	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
Segmentation	0.39	0.39	0.39	0.40	0.40	0.39	0.40	0.39	0.47	**0.49**
Abalone	0.04	0.04	0.04	**0.05**	**0.05**	**0.05**	**0.05**	0.04	0.04	0.04
Mushroom	0.12	0.12	0.12	0.16	0.15	0.13	0.13	0.12	0.14	**0.24**
PenDigits	0.41	0.41	0.41	0.42	0.42	0.43	0.44	0.41	**0.46**	**0.46**
Waveform	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**

TABLE C.5 – Les valeurs de l'ALC-ARI pour RN-BG

C.3 Les aires sous les courbes de MSE (ALC-MSE)

Le tableau C.6 (respectivement le tableau C.7) présente les performances moyennes en termes d'ALC-MSE de l'algorithme des K-moyennes standard précédé par Conditional Info (respectivement Rank Normalization) et en utilisant les différentes méthodes d'initialisation. Les résultats présentés dans les deux tableaux (C.6 et C.7) sont obtenus lorsque l'algorithme des K-moyennes est exécuté une, dix et cent fois.

R=1										
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**CD**	**K++R**	**SB**	**RS**
Iris	4.82	4.84	2.74	**2.63**	2.82	2.87	3.28	2.97	2.82	2.86
Hepatitis	0.30	0.30	0.23	**0.22**	0.24	0.30	0.24	0.24	0.23	0.25
Wine	5.99	5.99	6.06	5.88	6.03	6.15	**5.74**	6.27	6.52	6.45
Glass	12.28	12.32	**10.64**	11.40	11.69	13.97	11.76	10.93	11.50	11.81
Heart	0.21	0.21	**0.19**	**0.19**	**0.19**	0.20	**0.19**	**0.19**	**0.19**	0.20
Horsecolic	1.80	2.03	1.41	**1.27**	1.88	2.31	2.03	1.57	1.54	1.64
Soybean	34.67	34.74	28.91	29.55	26.22	27.38	24.93	24.24	**23.77**	23.99
Breast	0.85	0.87	0.81	0.82	0.84	0.86	**0.80**	0.82	0.83	0.85
Australian	0.22	0.22	**0.20**	**0.20**	0.21	0.21	**0.20**	**0.20**	**0.20**	0.21
Pima	0.27	0.28	0.24	**0.23**	0.25	0.28	0.25	0.25	**0.23**	0.25
Vehicle	3.70	3.68	3.34	3.42	3.81	4.31	3.47	**3.30**	3.32	3.36
Tictactoe	0.02	0.02	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
LED	3.17	3.23	2.43	2.50	2.81	3.21	2.59	**2.26**	2.28	2.31
German	0.29	0.29	**0.16**	0.17	0.18	0.23	0.19	0.17	**0.16**	0.18
Segmentation	22.71	22.77	22.15	22.60	**21.78**	21.97	22.01	22.28	23.40	23.42
Abalone	27.99	28.44	**20.99**	25.69	28.42	29.59	35.93	25.25	25.41	25.91
Mushroom	6.65	6.65	6.41	6.63	8.00	9.67	**6.17**	6.46	6.53	6.49
PenDigits	39.26	39.27	38.92	39.39	39.16	39.72	40.68	**38.90**	39.09	39.48
Waveform	1.33	1.33	**1.32**	1.33	1.34	1.39	1.33	**1.32**	1.34	1.34

R=10										
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**CD**	**K++R**	**SB**	**RS**
Iris	2.71	2.69	2.65	**2.63**	2.65	2.87	3.28	2.74	2.82	2.86
Hepatitis	0.24	0.24	**0.21**	0.22	0.23	0.30	0.24	0.22	0.23	0.25
Wine	6.00	6.02	6.00	5.89	5.89	6.15	**5.73**	6.06	6.51	6.45
Glass	10.70	10.74	**10.03**	11.40	10.92	13.97	11.76	10.24	11.49	11.81
Heart	0.19	0.20	**0.18**	0.19	0.19	0.20	**0.18**	**0.18**	0.19	0.20
Horsecolic	1.31	1.35	**1.23**	1.27	1.80	2.31	2.03	1.43	1.54	1.64
Soybean	29.04	29.33	24.28	29.51	23.95	28.02	24.75	**23.65**	23.73	23.99
Breast	0.80	0.81	**0.79**	0.82	0.81	0.86	0.80	**0.79**	0.82	0.85
Australian	0.20	0.20	**0.19**	0.20	0.20	0.21	**0.19**	**0.19**	0.20	0.21
Pima	0.23	0.23	**0.21**	0.23	0.23	0.28	0.24	0.22	0.23	0.25
Vehicle	3.14	3.15	**3.05**	3.42	3.43	4.31	3.45	3.06	3.31	3.36
Tictactoe	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
LED	2.56	2.63	2.22	2.50	2.59	3.20	2.58	**2.19**	2.26	2.31
German	0.17	0.17	**0.14**	0.17	0.16	0.23	0.19	0.15	0.16	0.18
Segmentation	21.74	21.74	21.51	22.60	**21.34**	21.97	21.96	21.65	23.29	23.42
Abalone	24.35	24.54	**19.60**	25.69	26.17	29.54	35.60	23.51	25.20	25.95
Mushroom	5.71	5.71	**5.67**	6.63	6.91	9.67	6.17	5.77	6.42	6.49
PenDigits	38.39	38.39	**38.28**	39.39	38.64	39.72	40.66	38.29	39.03	39.48
Waveform	**1.31**	**1.31**	**1.31**	1.33	1.32	1.39	1.33	**1.31**	1.34	1.34

R=100										
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**CD**	**K++R**	**SB**	**RS**
Iris	2.65	2.65	2.65	**2.63**	2.64	2.87	3.28	2.74	2.82	2.86
Hepatitis	0.22	0.22	**0.21**	0.22	0.22	0.30	0.24	**0.21**	0.23	0.25
Wine	5.98	6.01	5.95	5.88	5.83	6.15	**5.70**	5.95	6.49	6.45
Glass	10.36	10.41	**9.86**	11.40	10.74	13.97	11.75	10.02	11.45	11.81
Heart	0.19	0.19	**0.18**	0.19	**0.18**	0.20	**0.18**	**0.18**	**0.18**	0.20
Horsecolic	1.25	1.26	**1.21**	1.27	1.75	2.31	2.01	1.37	1.52	1.64
Soybean	27.07	27.24	**23.20**	28.13	23.61	27.38	24.75	23.28	23.71	23.99
Breast	0.78	0.78	**0.77**	0.82	0.79	0.86	0.80	**0.77**	0.82	0.85
Australian	0.20	0.20	**0.19**	0.20	**0.19**	0.21	**0.19**	**0.19**	0.20	0.21
Pima	0.21	0.21	**0.20**	0.23	0.22	0.28	0.24	0.21	0.23	0.25
Vehicle	3.03	3.03	**2.99**	3.42	3.31	4.31	3.43	3.00	3.31	3.36
Tictactoe	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
LED	2.38	2.42	**2.14**	2.50	**2.41**	3.19	2.55	2.14	2.25	2.31
German	0.14	0.14	**0.13**	0.17	0.16	0.23	0.19	0.15	0.16	0.18
Segmentation	21.45	21.46	21.30	22.60	**21.11**	21.97	21.94	21.39	23.20	23.42
Abalone	22.83	23.09	**18.98**	25.71	25.07	29.60	35.60	22.57	25.07	26.06
Mushroom	5.59	5.59	5.59	6.63	6.35	9.67	6.17	**5.62**	6.43	6.49
PenDigits	38.12	38.13	**38.06**	39.39	38.37	39.72	40.64	38.09	38.97	39.48
Waveform	**1.31**	**1.31**	**1.31**	1.33	1.32	1.39	1.33	**1.31**	1.33	1.34

TABLE C.6 – Les valeurs de l'ALC-MSE pour CI

R=1										
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**CD**	**K++R**	**SB**	**RS**
Iris	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
Hepatitis	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**
Wine	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
Glass	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
Heart	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**
Horsecolic	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	0.11	0.11	0.11	0.11	0.11
Soybean	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**
Breast	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
Australian	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**
Pima	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
Vehicle	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**
Tictactoe	0.11	0.11	0.11	0.11	0.11	**0.10**	**0.10**	0.11	**0.10**	0.11
LED	0.02	0.02	**0.01**	**0.01**	0.02	0.02	**0.01**	**0.01**	**0.01**	**0.01**
German	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
Segmentation	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**
Abalone	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Mushroom	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	0.06	**0.05**	**0.05**	**0.05**	0.06
PenDigits	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**
Waveform	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**

R=10										
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**CD**	**K++R**	**SB**	**RS**
Iris	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
Hepatitis	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**
Wine	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
Glass	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
Heart	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**
Horsecolic	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	0.11	0.11	0.11	0.11	0.11
Soybean	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**
Breast	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
Australian	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**
Pima	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
Vehicle	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**
Tictactoe	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	0.11
LED	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
German	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
Segmentation	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**
Abalone	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Mushroom	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	0.06	**0.05**	**0.05**	**0.05**	0.06
PenDigits	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**
Waveform	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**

R=100										
Données	**Sample**	**Random**	**K++**	**Var-Part**	**MM (Rand)**	**MM**	**CD**	**K++R**	**SB**	**RS**
Iris	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
Hepatitis	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**
Wine	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
Glass	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
Heart	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**
Horsecolic	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	0.11	0.11	0.11	0.11	0.11
Soybean	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**
Breast	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
Australian	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**
Pima	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
Vehicle	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**
Tictactoe	**0.10**	**0.10**	**0.10**	0.11	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	0.11
LED	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
German	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
Segmentation	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**
Abalone	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.01**	**0.00**	**0.00**	**0.00**
Mushroom	**0.05**	**0.05**	**0.05**	**0.05**	**0.05**	0.06	**0.05**	**0.05**	**0.05**	0.06
PenDigits	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**
Waveform	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**

TABLE C.7 – Les valeurs de l'ALC-MSE pour RN-BGB

Annexe D

Chapitre 5

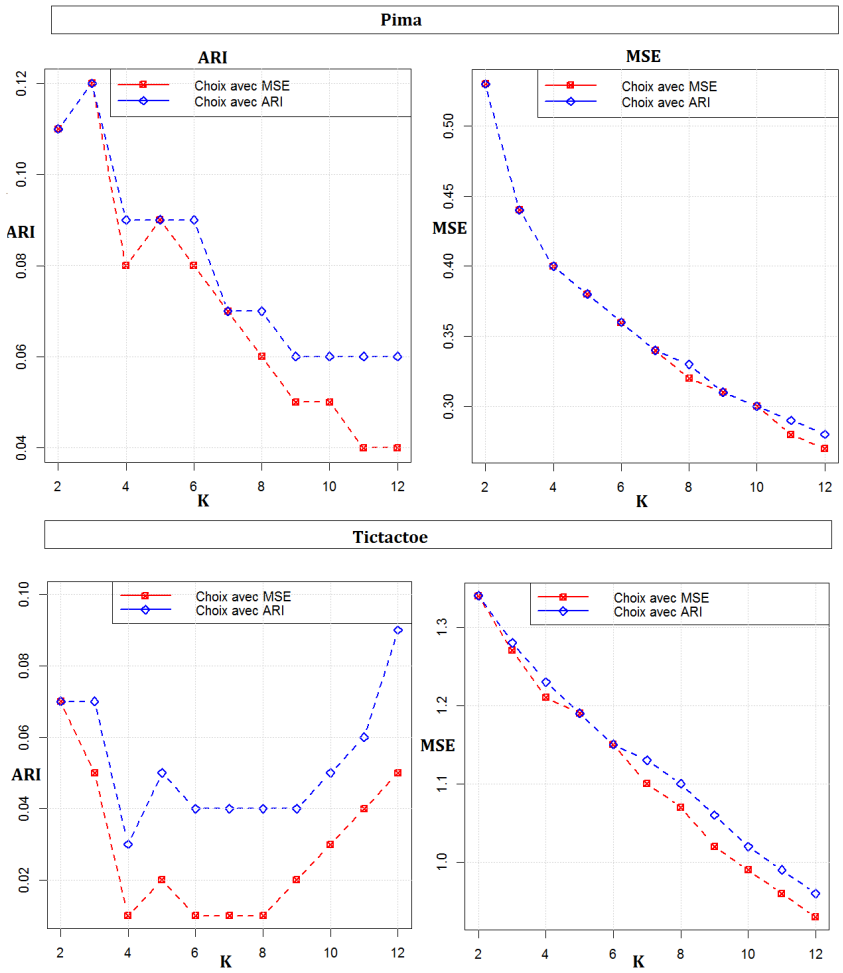


FIGURE D.1 – L'évolution des courbes de l'ARI (partie gauche) et de la MSE (partie droite) selon le critère utilisé pour choisir la meilleure partition pour les deux jeux de données Pima et Tictactoe

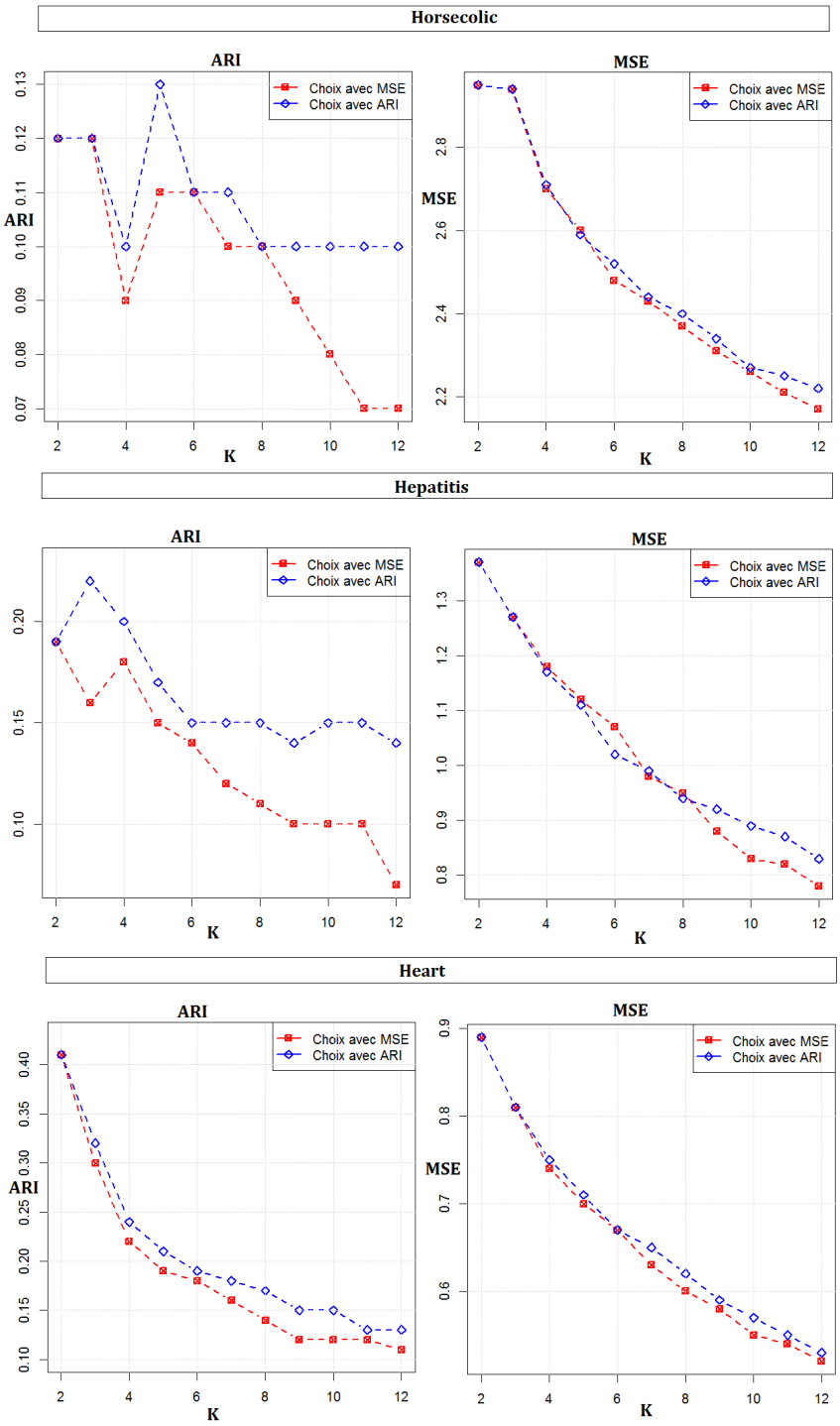


FIGURE D.2 – L'évolution des courbes de l'ARI (partie gauche) et de la MSE (partie droite) selon le critère utilisé pour choisir la meilleure partition pour les deux jeux de données Tictactoe

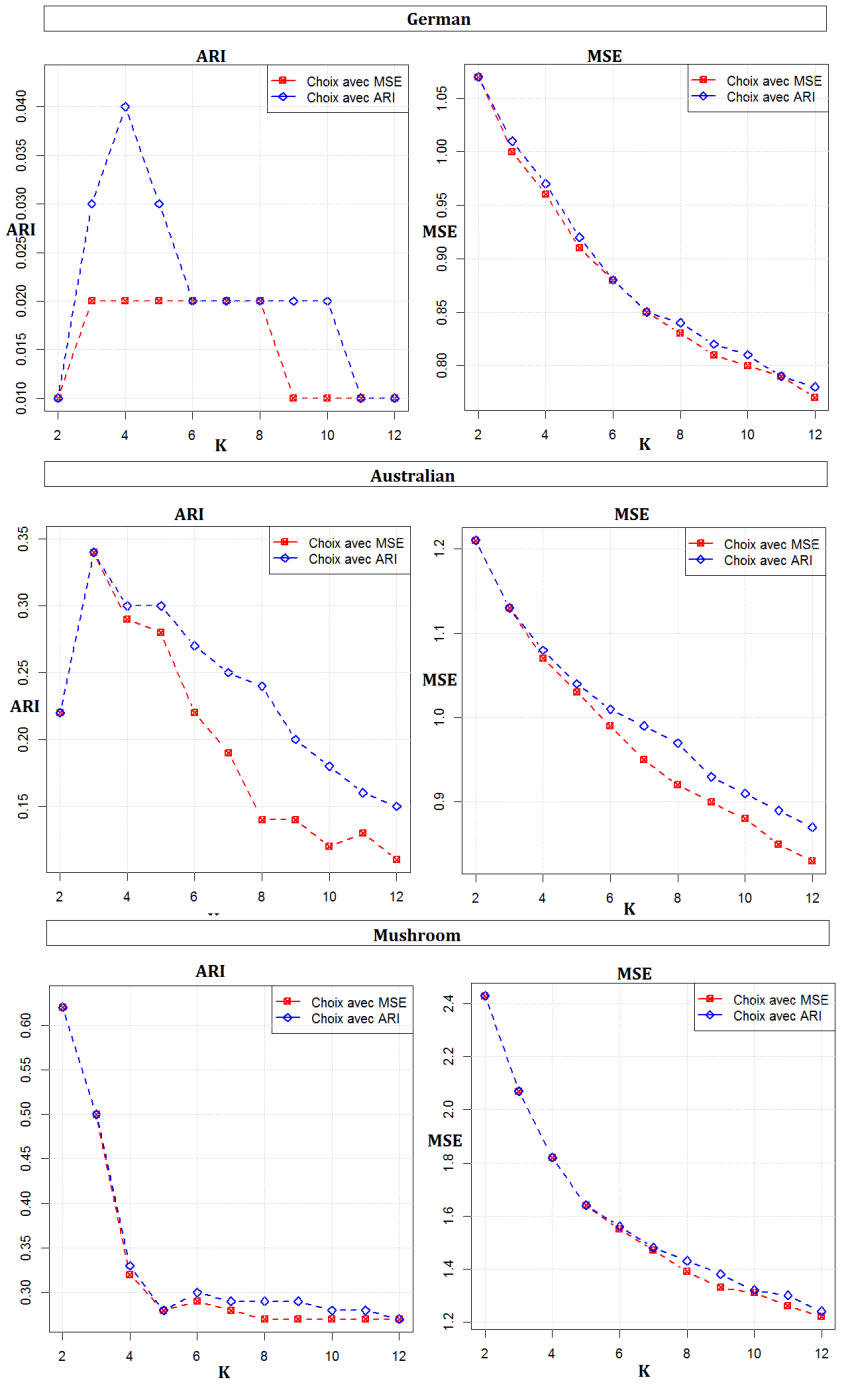


FIGURE D.3 – L'évolution des courbes de l'ARI (partie gauche) et de la MSE (partie droite) selon le critère utilisé pour choisir la meilleure partition pour les deux jeux de données Tictactoe

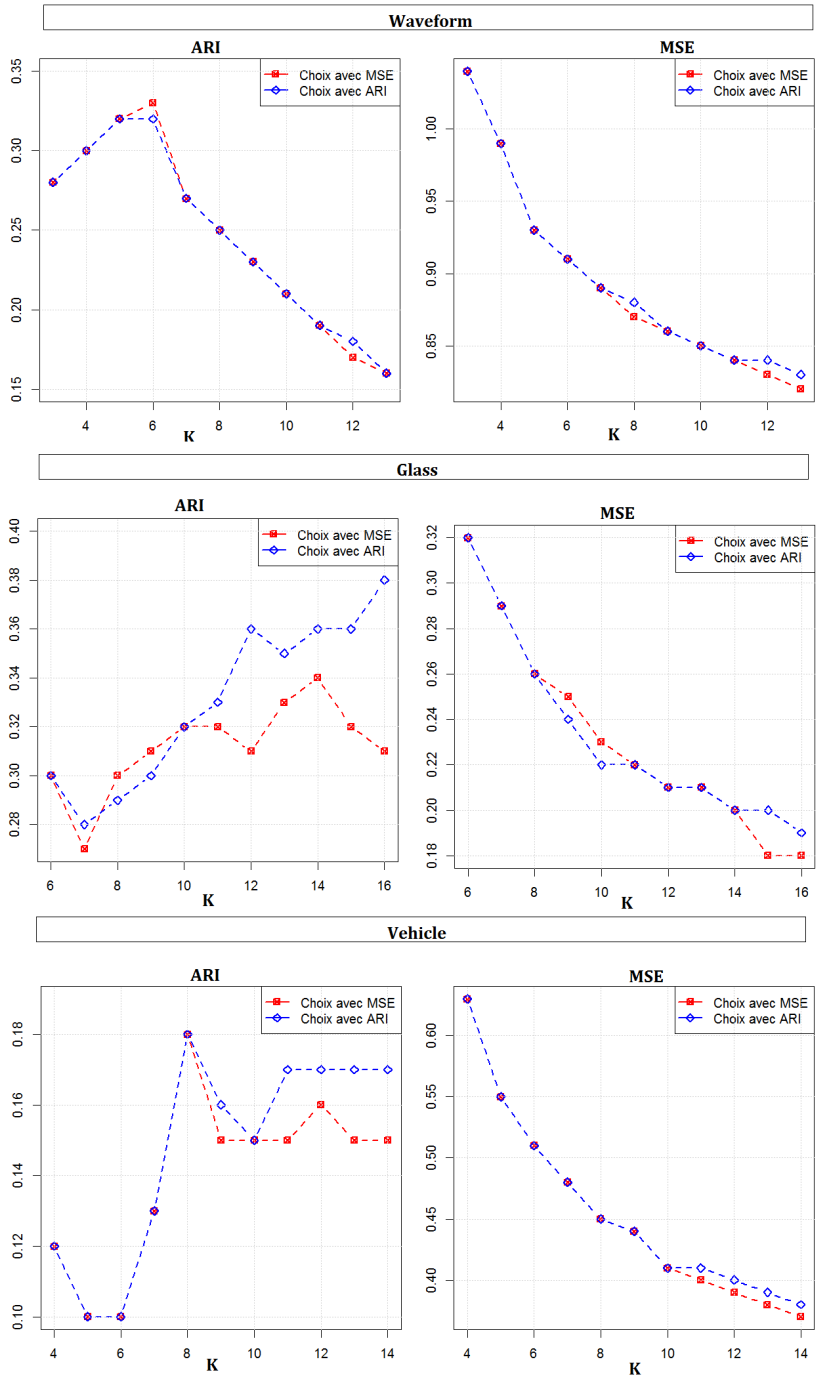


FIGURE D.4 – L'évolution des courbes de l'ARI (partie gauche) et de la MSE (partie droite) selon le critère utilisé pour choisir la meilleure partition pour les deux jeux de données Tictactoe

Annexe E

Mesure d'importance des variables

A Supervised Methodology to Measure the Variables Contribution to a Clustering

Oumaima Alaoui Ismaili[1,2], Vincent Lemaire[2], and Antoine Cornuéjols[1]

[1] AgroParisTech 16, rue Claude Bernard 75005 Paris
[2] Orange Labs, 2 av. Pierre Marzin, 22300 Lannion

Abstract. This article proposes a supervised approach to evaluate the contribution of explanatory variables to a clustering. The main idea is to learn to predict the instance membership to the clusters using each individual variable. All variables are then sorted with respect to their predictive power, which is measured using two evaluation criteria, i.e. accuracy (*ACC*) or Adjusted Rand Index (*ARI*). Once the relevant variables which contribute to the clustering discrimination have been determined, we filter out the redundant ones thanks to a supervised method. The aim of this work is to help end-users to easily understand a clustering of high-dimensional data. Experimental results show that our proposed method is competitive with existing methods from the literature.

1 Introduction

Everyday, huge amounts of data are generated by users via the web, social networks, etc. Clustering algorithms are a tool of choice to explore these high-dimensional data sets. However, their use is often hampered by the lack of understandability of the results. End-users would like to identify the most relevant variables that suffice to explain the observed clusters, but these are not easily detectable once a clustering has been performed. It is therefore crucial to be able to evaluate the contribution of each descriptive variable to the clustering process. Indeed, not all variables are relevant to the clustering: some may be irrelevant, some may be noisy and some may be redundant or (and) correlated.

The purpose of this study is to find a simple way to assist the analysts in their interpretation of a clustering result. The idea is to sort variables according to their contribution to a clustering using a supervised approach. The importance of a variable is evaluated as its power to predict the membership of each object to a cluster. In this paper, we restrict ourselves to an univariate classifiers to obtain an univariate weight for each variable.

The paper is organized as follows: Section 2 describes briefly some related work. Then, Section 3 presents the proposed method to score the contribution of variables to a clustering. This section also presents an alternative method to eliminate redundant variables among the relevant variables. The experimental results are presented Section 4. Finally, the perspectives and the further research are presented as a conclusion in the last section.

2 Related Work

Recently, the measure of the importance of the variables has been increasingly studied in the unsupervised learning. The methods proposed in this context can mainly be divided into two categories: *features selection* and *validation indices*.

C.K. Loo et al. (Eds.): ICONIP 2014, Part I, LNCS 8834, pp. 159–166, 2014.

Features selection methods can be grouped either as *wrapper* or as *filter* approaches. The *wrapper* approach aims to incorporate the feature selection in the clustering process, whereas, the idea of the *filter* approach is first to pre-select the features and then to use the selected features in the clutering process. In the unsupervised context, the *wrapper* methodology was initially proposed by Brodley in [1].

Inspired by the idea given in [1], Zhu et al. presented in [2] a novel method called ULAC. This method is essentially based on the analysis of the correlation among the variables. Moreover, some methods aim at removing the redundancy among variables. Accordingly, they rely on estimations of mutual information or of correlation ([3],[4],[5]). Mitra et al. proposed in [3] a method based on a measure of similarity between variables after elimination of the redundant variables. This measure is defined as the lowest eigenvalue of the correlation matrix. In [4], Vesanto et al. used a visualization tool (SOM-based approach) to detect the correlation between features. The same approach is used by Guerif et al in [5]. The difference between the two approaches is that Guerif et al. integrate a weight criterion in the SOM algorithm to reduce the effect of redundancy.

Other approaches have been presented to evaluate the clustering performance introducing criteria such as validation indices which can be adapted to evaluate the variables importance. Those approaches are divided in two main types: *external* and *internal* [6]. The *external* approaches exploit the supervised information given by the ID-cluster (identification given to each discovered cluster that can be subsequently used as a "label"). Among these approaches, we can cite: Adjusted Rand index [7], F-measure [8] and MMI [9]. The internal approaches use unsupervised criteria like the inertia. Among these methods, we can cite: Davies-Bouldin [10], Silhouette [11], Dunn-index [12], SD [13], XB-index [14], I-index [14] and BIC [15] indices.

3 Contribution

In this section, we propose two supervised approaches which fall within the context of the external validation indices. These approaches allow an interpretation of the clustering output based on relevant variables in case where the clsutering does not suffer from a very bad quality (otherwise there is no sense to interpret the result). In the remainder of this paper, we call this output (or the clustering result) *'the reference clustering'*. The first supervised approach consists in measuring the variables importance with respect to their predictive power regarding the cluster Ids. The second one aims at detecting the redundant variables.

3.1 Variables Importance

The objective of this work is to propose a simple way to identify the most relevant features from the output of a clustering. In order to retain all variables, we rank the variables according to their importance without doing a selection. The main idea is to

turn this problem into a supervised classification problem where the cluster membership (ID-cluster) is used as a target class. Then, for each variable, we use a supervised classification algorithm to predict the ID-cluster. We define the importance of variables as their power to predict the ID cluster: a variable is relevant only if it is able to predict correctly the ID cluster obtained from the reference clustering (i.e. clustering using all variables). To measure the importance of each variable, we use two evaluation criteria: Accuracy and Adjusted Rand Index:

- *Accuracy* (ACC) criterion: a variable is considered relevant if the associated accuracy value is high.
- *Adjusted rand index (or ARI)* is a popular cluster validation index proposed by Hubert and Arabie [7]. It can be used to evaluate the performance of the classification as in [16]. In this work, we calculate the ARI between: (i) the reference clustering (ii) the predicted membership (ID-cluster) associated to the variable of which we want to measure the importance. The idea behind this is to compare the reference clustering with each predictive membership associated to each variable. So, a variable is important if the associated predictive ID-cluster is highly similar to the reference clustering, i.e. the ARI value is close to 1.

The algorithm 1 presented below provides a summary of our approach:

An interesting measure of importance must allow us to sort variables according to their relevance in a clustering process and the least influent variables should only contain little or irrelevant information to create the clusters. Consequently, the quality of the obtained clustering which is deprived of these variables remains substantially the same or even slightly better (less noise). In contrast, the removal of an important variable deprives the algorithm of important information and leads to a poor clustering result.

Notations:
X: The training database constituted of N examples and d explanatory variables, (X_{ab} is the value of the variable b for the example a)
M: A supervised classifier
CLU: A clustering algorithm
M_{ref}: The reference clustering model
$IdClusters$: A vector of the N memberships
R : Ranking of the d explanatory variables

$XPRE \leftarrow$ `preprocessing` (X)
$M_{ref} \leftarrow$ `train` (CLU, XPRE)
$IdClusters \leftarrow$ Membership($XPRE$, M_{ref})
for *i=1 to d* **do**
  $M_i \leftarrow$ `train` ($XPRE_{.i}$, $IdClusters$)
  $ACC_i \leftarrow$ `computeAccuracy` (M_i)
  $ARI_i \leftarrow$ `computeAdjustedRandIndex` (M_i)
end
$R_{ACC} \leftarrow$ `sortInDescendingOrder` (ACC_i, i=1 to d)
$R_{ARI} \leftarrow$ `sortInDescendingOrder` (ARI_i, i=1 to d)

Algorithm 1. Algorithm for ranking

To compare our proposed method to other existing methods from the literature, the curve of the ARI values versus the number of variables used will be plotted. This curve is obtained as follows:

For each iteration until one reaches the number of variables:

- Eliminate the less relevant variable with respect to the chosen criterion;
- New partition: run the clustering algorithm without this variable;
- Calculate the ARI value between the reference clustering and the new partition.

The review of the results can be visually made by observing the curve evolution (for example, see Figure 1).

3.2 Redundant Variables

Once the variables that are the most informative for the clustering have been identified, it is important to filter out the redundant ones in order to improve the understandability of the result. To solve this problem, we propose a supervised approach.

The concept of redundancy is based on the similarity between partitions obtained using the "predicted ID-Clusters" (using Algorithm 1) for each variable. The assumption is: X_i and X_j are redundant if they produce similar partitions when considering their "predicted ID-Clusters" (using M_i and M_j). A way to measure the similarity between these two partitions is to use the ARI criterion. For example, the ARI criterion will be close to 1 when it calculated between two partitions containing same "predicted ID-Clusters" or between two partitions containing symetric "predicted ID-Clusters". The resulting algorithm is presented below (see Algorithm 2).

Notations:
X: The training database constituted of N examples and d explanatory variables
$M_.$: d supervised classifier models coming from the Algorithm 1
$PredId$: A vector of size N of the predicted ID-Cluster for a given explanatory variable
RE: A matrix of size dxd values

$XPRE \leftarrow$ `preprocessing` (X)
for *i=1 to d* **do**
  $PredId(d) \leftarrow$ `PredictionOfTheMembership` (M_i, $XPRE_{.i}$)
end
for *all pairs of variable (l, m)* **do**
  $RE(l, m) \leftarrow$ `computeAdjustedRandIndex` ($PredId(l)$,$PredId(m)$)
end

Algorithm 2. Algorithm for redundant variables

4 Experimental Results

4.1 Protocol

To evaluate the behavior of our approach, we have selected 3 different datasets from the UCI [17]: WINE, PIMA and WAVEFORM datasets. The two first datasets are used to

illustrate the competitiveness of the proposed method to measure the variables importance comparing to two other methods from the literature. Among these methods, we decide to use efficient and often used indexes from the literature: Davies-Bouldin [10] and SD indexes [13]. The last dataset is used to illustrate the behavior of our approach to detect the redundant variables.

We proceed as follows to evaluate the performance of our approach:

- the pre-processing used is standardization[1];
- to obtain the reference clustering, the K-means algorithm [18] has been used where:
  - K is equal to the number of target class for each used datasets (as in [19]);
  - the method used to initialize the centroids is K-means++ algorithm [20];
  - the number of replicates is 25 [2].
- a decision tree (CART) [21] has been used to predict the ID-cluster[3].

4.2 Variables Contribution

The first experimentation to test our approach is made using the WINE dataset which is constituted of $N = 178$ sample points described with $d = 13$ variables and associated with three different classes. The ARI obtained between the reference clustering (using K-means algorithm, where K=3) and the target class is equal to 0.91. Figure 1 presents the evolution of the ARI curve for the three approaches (SD, DB and the supervised approach using ARI or ACC to measure the contribution of variables in the clustering results) versus the number of variables. The table 1 (left part) presents the list of the ranked variables (from the most important to the least important) for the three approaches.

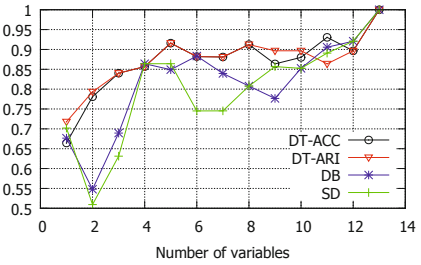


Fig. 1. Evolution of the ARI criterion for the 4 methods (K=3)

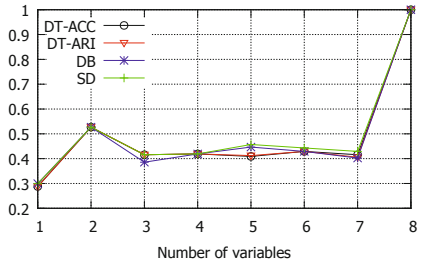


Fig. 2. Evolution of the ARI criterion for the four methods (K=2)

[1] All the experimentation have been realized using R (`http://www.r-project.org/`) and are easily reproducible.

[2] The initialization process and the nature of the K-means algorithm does not guarantee to reach a global minimum. Therefore the algorithm has to be run several times.

[3] To evaluate the importance of the variables for the clustering, we need to choose a classifier which does not modify the representation used to elaborate the reference clustering; i.e the data after the pre-processing step.

Table 1. Ranking of the variables

Index	Wine													Pima							
DB	V7	V6	V10	V1	V12	V13	V9	V8	V4	V5	V3	V2	V11	V8	V2	V1	V3	V6	V5	V4	V7
SD	V7	V6	V10	V1	V12	V9	V8	V13	V5	V4	V3	V2	V11	V8	V2	V1	V3	V6	V7	V4	V5
ARI-Tree	V7	V13	V12	V1	V10	V6	V11	V2	V9	V4	V8	V5	V3	V8	V2	V1	V3	V5	V6	V4	V7
ACC-Tree	V7	V13	V12	V1	V10	V6	V11	V2	V9	V4	V5	V8	V3	V8	V2	V1	V3	V5	V6	V4	V7

The PIMA data dataset contains $N = 768$ sample points described with $d = 8$ variables which are associated with two different classes. The ARI obtained between the reference clustering (using K-means algorithm, where $K = 2$) and the target class is equal to 0.11. Table 1 (right part) and Figure 2 present respectively the list of the ranked variables (from the most important to the least important) and the evolution of ARI curve for the three approaches (DB, SD and the proposed approach).

The results obtained on PIMA and WINE show that the proposed method is competitive with regards to DB and SD approaches on these two datasets.

4.3 Redundant Variables

To test the ability of our approach to detect the redundant variables, we use the WAVEFORM dataset. This dataset consists of $n = 5000$ sample points described with 40 variables and associated with three different classes: only the first 21 variables are real attributes for this database and most of these are relevant to a classification problem whereas the last 19 variables are noisy standard centered Gaussian variables (for more details see [21], page 43 - 49). Figure 3 shows that the proposed method identifies the irrelevant set of variables $W = V1, V21 - V37, V39, V40$. The remaining variables are all relevant variables for the clustering.

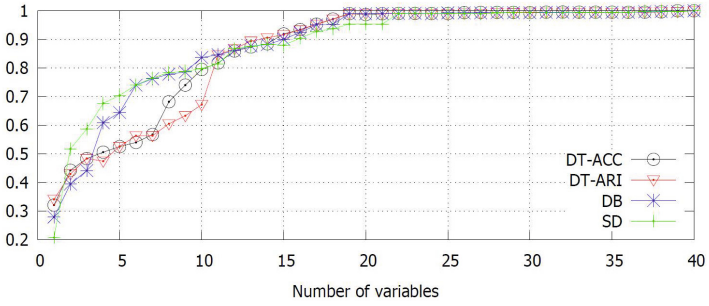


Fig. 3. Evolution of ARI criterion for the three methods (K=3)

To identify the redundant variables, we use the method described in Section 3.2. Table 2 presents the ARI values calculated between two pairs of relevant variables (the 20 variables identified by the proposed method using the ACC criterion). In this table, if we consider only the values above 0.96 to focus the attention on the high values of redundancy. The set of redundant variables is then : $R = V38, V2, V20, V19, V3$. Finally the set of relevant variables is $V = V4 - V18$. These obtained results are similar to those obtained using RD-MCM selection features method (see [19]). The ARI value obtained between the predicted ID-cluster using all variables (41 variables) and the predicted ID-cluster using the relevant variable (18 variables) is equal to 0.935.

Table 2. ARI values between pairs of relevant variables

	V7	V15	V8	V14	V16	V6	V13	V12	V17	V9	V5	V10	V4	V18	V11	V3	V19	V20	V2	V38
V7	1,00	0,51	0,42	0,42	0,39	0,41	0,41	0,38	0,37	0,36	0,35	0,34	0,34	0,34	0,32	0,32	0,32	0,32	0,32	0,32
V15		1,00	0,66	0,62	0,59	0,59	0,58	0,54	0,52	0,50	0,49	0,46	0,46	0,46	0,44	0,44	0,44	0,44	0,44	0,44
V8			1,00	0,76	0,72	0,70	0,67	0,62	0,59	0,56	0,54	0,51	0,51	0,51	0,49	0,48	0,48	0,48	0,48	0,48
V14				1,00	0,81	0,78	0,75	0,67	0,62	0,59	0,57	0,54	0,53	0,53	0,51	0,51	0,50	0,51	0,50	0,50
V16					1,00	0,84	0,77	0,71	0,66	0,61	0,60	0,56	0,56	0,56	0,53	0,53	0,52	0,53	0,52	0,52
V6						1,00	0,86	0,75	0,68	0,63	0,62	0,58	0,57	0,58	0,55	0,54	0,54	0,54	0,54	0,54
V13							1,00	0,79	0,72	0,66	0,65	0,60	0,60	0,60	0,58	0,57	0,57	0,57	0,57	0,57
V12								1,00	0,88	0,81	0,78	0,74	0,73	0,73	0,69	0,69	0,68	0,68	0,68	0,68
V17									1,00	0,89	0,84	0,81	0,79	0,79	0,75	0,74	0,74	0,74	0,74	0,74
V9										1,00	0,91	0,86	0,85	0,84	0,80	0,79	0,79	0,79	0,79	0,79
V5											1,00	0,89	0,87	0,86	0,83	0,82	0,81	0,82	0,82	0,82
V10												1,00	0,94	0,91	0,88	0,87	0,86	0,86	0,86	0,86
V4													1,00	0,94	0,89	0,88	0,87	0,87	0,87	0,87
V18														1,00	0,92	0,89	0,88	0,88	0,89	0,88
V11															1,00	0,95	0,93	0,92	0,92	0,92
V3																1,00	**0,96**	0,95	0,93	0,94
V19																	1,00	**0,97**	0,95	**0,96**
V20																		1,00	**0,97**	**0,97**
V2																			1,00	**1,00**
V38																				1,00

5 Conclusion

This paper has presented a supervised method to measure the importance of the variables used in a clustering. This method turned the problem into a supervised classification problem to sort variables according to their importance at the end of the clustering convergence. The experimental results corroborated the competitiveness of the method comparing to other methods from the literature. It has been incorporated successfully in the process of marketing service in the french Orange company. Future works will be done to incorporate the method in the convergence of the clustering algorithm and to measure the variables importance as a multivariate supervised classification problem.

References

1. Dy, J.G., Brodley, C.E.: Feature selection for unsupervised learning. J. Mach. Learn. Res. 5, 845–889 (2004)
2. Liu, P., Zhu, J., Liu, L., Li, Y., Zhang, X.: Application of feature selection for unsupervised learning in prosecutors' office. In: Wang, L., Jin, Y. (eds.) FSKD 2005. LNCS (LNAI), vol. 3614, pp. 35–38. Springer, Heidelberg (2005)

3. Mitra, P., Murthy, C.A., Pal, S.K.: Unsupervised feature selection using feature similarity. IEEE Trans. Pattern Anal. Mach. Intell. 24(3), 301–312 (2002)
4. Vesanto, J., Ahola, J.: Hunting for correlations in data using the self-organizing map. In: Proceeding of the International ICSC Congress on Computational Intelligence Methods and Applications (CIMA 1999), pp. 279–285. ICSC Academic Press (1999)
5. Guérif, S., Bennani, Y., Janvier, E.: μ-SOM: Weighting features during clustering. In: Proceeding of the 5th Workshop on Self-organizing Maps (WSOM 2005), pp. 397–404 (2005)
6. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. J. Intell. Inf. Syst. 17(2-3), 107–145 (2001)
7. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification 2(1), 193–218 (1985)
8. Larsen, B., Aone, C.: Fast and effective text mining using linear-time document clustering. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ICDM), pp. 16–22. ACM, New York (1999)
9. Alok, A.K., Sriparna, S., Ekbal, A.: A min-max distance based external cluster validity index: MMI. In: HIS, pp. 354–359. IEEE (2012)
10. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1(2), 224–227 (1979)
11. Rousseeuw, P.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. (1), 53–65 (1987)
12. Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics 3(3), 32–57 (1973)
13. Halkidi, M., Vazirgiannis, M., Batistakis, Y.: Quality scheme assessment in the clustering process. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 265–276. Springer, Heidelberg (2000)
14. Xie, X.L., Beni, G.: A validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell. 13(8), 841–847 (1991)
15. Raftery, A.: A note on Bayes factors for log-linear contingency table models with vague prior information. Journal of the Royal Statistical Society, 249–250 (1986)
16. Santos, J.M., Embrechts, M.: On the use of the adjusted rand index as a metric for evaluating supervised classification. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009, Part II. LNCS, vol. 5769, pp. 175–184. Springer, Heidelberg (2009)
17. Blake, C.L., Merz, C.J.: Uci repository of machine learning databases (1998)
18. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Cam, L.M.L., Neyman, J. (eds.) Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press (1967)
19. Celeux, G., Martin-Magniette, M.L., Maugis, C., Raftery, A.E.: Comparing model selection and regularization approaches to variable selection in model-based clustering. Journal de la Société Française de Statistique (2014)
20. Arthur, D., Vassilvitskii, S.: K-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, pp. 1027–1035 (2007)
21. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and regression trees. Wadsworth International Group (1984)

Une méthode basée sur des effectifs pour calculer la contribution des variables à un clustering

Oumaima Alaoui Ismaili[*,***], Julien Salotti[**], Vincent Lemaire[***]

[*]AgroParisTech 16, rue Claude Bernard 75005 Paris
[**]INSA Lyon 20, avenue Albert Einstein - 69621 Villeurbanne
[***]Orange Labs 2 avenue Pierre Marzin 22300 Lannion

Résumé. Cet article présente une étude préliminaire effectuée dans un contexte industriel. On dispose d'une typologie de clients que le service marketing souhaite contacter. Cette typologie est une segmentation des clients en groupes de clients dont les profils seront utilisés pour proposer des campagnes marketing différenciées. La constitution des groupes est réalisée à l'aide d'une technique de clustering qui ne permet pas actuellement de connaître l'importance des variables explicatives (qui décrivent les clients). Cet article propose de résoudre ce problème à l'aide d'une méthodologie qui donne dans notre contexte industriel, l'importance des variables explicatives. Cette méthode sera comparée à certaines méthodes de l'état de l'art.

1 Introduction

Lorsqu'on désire contacter un client pour lui proposer un produit on calcule au préalable la probabilité qu'il achète ce produit. Cette probabilité est calculée à l'aide d'un modèle prédictif pour un ensemble de clients. Le service marketing contacte ensuite ceux ayant les plus fortes probabilités d'acheter le produit. En parallèle, et avant le contact commercial, on réalise une typologie des clients auxquels on propose des campagnes différenciées par groupes. Plus formellement, le problème est celui du clustering supervisé, où un clustering est appliqué sur des données étiquetées. Ce problème peut être défini comme étant un processus de regroupement des individus en clusters, tels que les données de chaque cluster soient les plus similaires possibles et appartiennent à la même classe à prédire. Le lecteur pourra trouver une description détaillée de ce problème dans l'article (Lemaire et al., 2012).

Actuellement, la technique de clustering utilisée pour la constitution des groupes ne permet pas d'identifier les variables les plus importantes. Autrement dit, cette technique ne permet pas de connaître les variables qui contribuent le plus lors de la construction des clusters. Par conséquent, le service marketing éprouve des difficultés à adapter sa campagne aux différents profils identifiés. L'objectif de cette étude est donc de proposer une méthode qui permet de mesurer l'importance des variables à la fin de la convergence d'un clustering. Cette dernière doit prendre en compte trois points principaux :

1. Conserver toutes les variables utilisées lors du clustering.
2. Ne pas réapprendre le modèle.

3. Garder l'espace de représentation des données utilisé lors de la phase de prétraitement (discrétisation pour les variables continues et groupage des valeurs pour les variables catégorielles) qui précède la phase de clustering.

Au vu du contexte d'étude, cet article propose de poser le problème de mesure de contribution des variables comme un problème de classification supervisée. C'est à dire apprendre à prédire l'appartenance aux clusters à partir d'une variable explicative donnée, puis d'ordonner les variables selon leur pouvoir prédictif.

La section 2 de cet article décrit la méthode de clustering utilisée qui contraint le problème de calcul d'importance. La section 3 décrit la solution proposée pour trier les variables en fonction de leur importance dans ce contexte. La section 4 présente des résultats préliminaires avant de conclure au cours de la dernière section.

2 L'existant : la méthode de clustering utilisée

L'ensemble des notations qui seront utilisées par la suite, sont les suivantes :

- Une base d'apprentissage, E, comportant N éléments (individus), M variables explicatives et une variable Y à prédire comportant J modalités (les classes à prédire sont C_j).
- Chaque élément D des données est un vecteur de valeurs (continues ou catégorielles) $D = (D_1, D_2, \ldots, D_M)$.
- K est utilisé pour désigner le nombre de classes souhaitées.

2.1 L'algorithme de clustering

L'algorithme de clustering utilisé est décrit dans (Lemaire et al., 2012). Cet article a montré que si on utilise un algorithme de type k-moyennes à l'aide d'une présentation supervisée et de la norme L1, on obtient des clusters où deux individus proches au sens de la distance seront proches au sens de leur probabilité d'appartenance à la classe cible (voir équation 5 dans (Lemaire et al., 2012)). Cet algorithme peut être présenté de la manière suivante :

- Prétraitements des données (voir section 2.2)
- Pour replicate=1 à R [1]
  - initialisation des centres (voir section 2.3)
  - Algorithme usuel des k-moyennes avec comme centre une approximation de la médiane (Kashima et al., 2008) et la norme L1 (Jajuga, 1987)
- choix de la meilleure "replicate" parmi les R solutions obtenues (voir section 2.4)
- présentation des résultats (voir section 2.5)

Cet algorithme est en partie supervisé puisque les prétraitements et le choix du meilleur "replicate" sont basés sur des critères supervisés qui sont décrits ci-dessous.

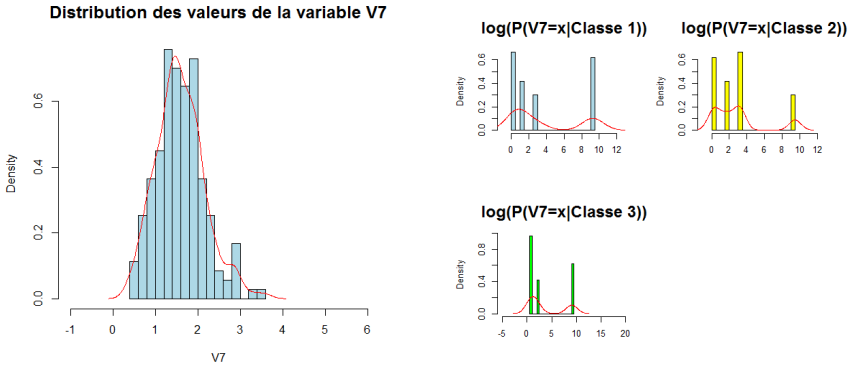
1. Dans cette étude, on fixe le nombre de replicates à R= 50

2.2 Représentation supervisée des données

Une représentation supervisée des données est utilisée. Elle recode les données brutes grâce à une technique de groupage supervisée ou de discrétisation supervisée qui utilise la variable *cible* contenant la liste des classes à prédire.

Les variables continues sont discrétisées (Boullé, 2004), c'est à dire découpées en intervalles, tandis qu'une méthode de groupage est appliquée sur les variables catégorielles (Boullé, 2005). Le prétraitement des données est réalisé à l'aide de l'approche MODL. Cette approche consiste à trouver la partition des valeurs de la variable continue (respectivement catégorielle) qui donne le maximum d'information sur la répartition des classes à prédire connaissant l'intervalle de discrétisation (respectivement le groupe de modalités).

A la fin du processus de prétraitement, les variables numériques et catégorielles sont donc recodées : chaque variable m est recodée en une variable qualitative contenant I_m valeurs de recodage. Chaque objet de données est alors recodé sous forme d'un vecteur de modalités discrètes $D = D_{1i_1}, D_{2i_2}, \ldots, D_{Mi_M}$. D_{mi_m} représente la valeur de recodage de D_m sur la variable m, avec la modalité discrète d'indice i_m. Ainsi les variables de départ sont alors toutes représentées sous une forme numérique. Le vecteur initial contenant M composantes de variables numériques et catégorielles devient un vecteur de $M * J$ composantes numériques : $log(P(D_{mi_m}|C_j))$.



(a) avant le prétraitement de la variable (b) après le prétraitement de la variable

FIG. 1 – *la distribution des valeurs de la variable V7 avant et après le prétraitement*

A titre illustratif, la figure 1 présente la discrétisation d'une variable numérique de la base UCI (Blake et Merz, 1998) "Wine" qui contient 13 variables explicatives et une variable cible à 3 classes ($Y \in \{1, 2, 3\}$). Après le prétraitement, on remarque que la distribution des variables prétraitées est multimodale et non gaussienne.

2.3 Initialisation des centres

L'initialisation des algorithmes de clustering basés sur le partitionnement influence la qualité de la solution trouvée et le temps d'exécution. C'est pourquoi le choix de la méthode d'initialisation de l'algorithme est un choix important lors de l'implémentation d'un algorithme de

clustering. Cependant, il n'y a pas une méthode d'initialisation meilleure que toutes les autres dans la littérature (Meila et Heckerman, 1998) mais plusieurs bonnes méthodes. Parmi ces dernières la méthode nommée K means++ a été utilisée (Arthur et Vassilvitskii, 2007). Cet algorithme est défini comme suit :

1. Choisir un centre uniformément au hasard parmi l'ensemble des points de données E.
2. Pour chaque point D, calculer $S(D)$: la distance entre D et le centre le plus proche qui a déjà été choisi.
3. Choisir le centre prochain $c_i = D' \in E$ suivant la probabilité $\frac{S(D')^2}{\sum_{D \in E} S(D)^2}$.
4. Répéter les étapes 2 et 3 jusqu'à ce que l'on ait placé tous les centres.

2.4 Choix de la meilleure replicate

Afin de prémunir contre le problème lié à l'initialisation et au fait que l'algorithme ne garantit pas d'avoir un minimum global, on exécute l'algorithme de clustering plusieurs fois. On obtient donc un certain nombre de partitionnements différents, dont on souhaite garder uniquement le meilleur. Pour se faire, et puisqu'on est dans le cadre de clustering supervisé, on utilise une mesure de qualité nommée *EVA* qui mesure la qualité d'un clustering supervisé en prenant en considération la variable 'cible'.

EVA mesure le gain qu'une partition établissant un compromis entre le nombre de groupe et la répartition des étiquettes peut apporter par rapport à la partition ayant un seul groupe. Plus formellement, EVA est une description scalaire comprise entre 0 et 1, décrite par la formule suivante : $EVA = 1 - \left(\frac{c(K)}{c(1)}\right)$, où

$$c(K) = log(N) + log\left(\begin{array}{c} N+K-1 \\ K \end{array}\right) + \sum_{k=1}^{K} log\left(\begin{array}{c} N_k+J-1 \\ J-1 \end{array}\right) + \sum_{k=1}^{K} log\left(\frac{N_k!}{N_{k1}! \ldots N_{kJ}!}\right) \quad (1)$$

et où K est le nombre de cluster, N_{kj} est le nombre d'individus du cluster k et de classe j et N_k le nombre d'individus dans le cluster k.

$c(K)$ mesure d'une manière supervisée l'intérêt d'une partition de Voronoi relative à un échantillon. Il quantifie le compromis entre le nombre de groupes de la partition et la distribution de la variable cible, ce qui correspond à un compromis entre complexité du modèle et ajustement du modèle aux données de l'échantillon. D'une manière générale, on cherche à maximiser cette mesure. Cette mesure est détaillée dans(Ferrandiz et Boullé, 2010).

2.5 Présentation des résultats du clustering

A la fin de la convergence de la méthode de clustering, on présente les résultats à l'aide des groupes de modalités et des intervalles créés lors de l'étape de prétraitement, en calculant les effectifs des individus dans chaque groupe de modalités ou intervalle pour chaque cluster. A titre d'exemple, le tableau 1 présente les effectifs des individus dans l'ensemble des intervalles de la variable V7 de la base Wine pour les trois clusters.

		id-cluster			
	Intervalle / Groupe de modalités	Cluster 1	Cluster 2	cluster 3	Total
V1	...	...	...	...	...
	...	...	...	...	...
	...	...	...	...	...
...					
V7	$]-\infty ; 0.975]$	0	1	38	39
	$]0.975 ; 1.575]$	0	13	10	23
	$]1.575 ; 2.31]$	1	38	0	39
	$]2.31 ; +\infty]$	58	19	0	77
...					
V13	...	...	...	...	...
	...	...	...	...	...
	...	...	...	...	...
	...	...	...	...	...

TAB. 1 – *Discrétisation de la variable V7*

3 Choix d'une méthode de tri adaptée au contexte

3.1 Contribution d'une variable

Dans la littérature, plusieurs indices de qualité de clustering ont été développés afin de mesurer la contribution d'une variable au résultat d'un clustering. Cette problématique de mesure de contribution, de mesure d'importance, dans un clustering peut être divisée en deux sous-problèmes que l'on peut respectivement caractériser de *global* ou *local*. L'importance *globale* a pour but de mesurer l'impact que la variable a eu sur la structure entière du partitionnement et non pas l'impact qu'elle a eu sur un cluster en particulier. Par contre, l'importance *locale* a pour objectif de savoir quelle variable a été déterminante dans la formation d'un cluster en particulier. Nous nous intéressons dans cet article uniquement à l'importance globale.

Parmi les méthodes de l'état de l'art permettant de mesurer cette importance on trouvera de nombreux indices tels que : (i) l'indice de Dunn (Dunn, 1974) ; (ii) l'indice de Davies-Bouldin (DB) (Davies et Bouldin, 1979) ; (iii) l'indice Silhouette (Rousseeuw, 1987) ; l'indice SD (Halkidi et al., 2000) ; l'indice S_Dbw(Halkidi et Vazirgiannis, 2001) ...

La plupart de ces méthodes utilisent le théorème de Huygens et la décomposition de l'inertie totale en la somme de l'inertie intra cluster et de l'inertie inter cluster. La contribution d'une variable est alors, par exemple, calculée en mesurant la valeur de l'inertie inter calculée uniquement avec cette variable vis-à-vis de la somme des inerties inter calculée sur toutes les variables ((Benzécri, 1983), (Celeux et al., 1989) section 2.10 p154-164).

3.2 Notre proposition

Notre but est l'ordonnancement du tableau 1 selon la contribution des variables à l'affection des clusters. Nous pensons que dans le cadre de notre contexte et de nos prétraitements les critères classiques tel que ceux présentés ci-dessus ne sont pas totalement adaptés. La figure 1 montre par exemple que pour la base de données Wine la distribution de départ de la variable V7 (partie gauche de la figure) devient après prétraitements « multimodale » (partie droite de la figure).

Nous décidons alors de poser le problème comme un problème de classification supervisée. Le but sera d'essayer d'apprendre à prédire le cluster d'appartenance d'un individu (l'id-cluster du tableau 1) en utilisant une seule variable (classification univariée). Puis de trier les variables selon leur pouvoir prédictif vis-à-vis de l'id-cluster.

Comme on désire trier les variables selon le résultat de clustering initialement obtenu on s'interdira les classifieurs qui créent une nouvelle représentation des données. En effet on ne souhaite pas mesurer l'importance des variables dans un nouvel espace mais l'importance des variables avec la représentation supervisée obtenue juste avant la création des clusters. Le but est l'aide à l'interprétation du clustering de manière à permettre à l'analyste de concentrer son attention sur les variables les plus importantes vis-à-vis du clustering obtenu.

Parmi les méthodes capables d'utiliser la représentation issue de nos prétraitements supervisés et le tableau d'effectifs qui sert à présenter les résultats on choisit d'utiliser la méthode MODL qui mesure le pouvoir prédictif (appelé "level") d'une variable numérique dans (Boullé, 2004) et le pouvoir prédictif d'une variable catégorielle dans (Boullé, 2005).

Dans le cas d'une variable numérique [respectivement catégorielle] si les intervalles de discrétisation [les groupes de modalités] sont fixés, alors le critère se calcul à l'aide des effectifs observés dans les intervalles [groupes de modalités]. Nos prétraitements supervisés nous donnent les intervalles [les groupes de modalités] et la projection des individus sur les clusters (tableau 1) nous permettent d'avoir en notre possession les effectifs. L'ensemble des éléments nécessaire au calcul du level par variable est donc disponible pour toutes les variables explicatives.

4 Expérimentations

4.1 Jeu de données utilisé

Pour évaluer le comportement de notre nouvelle approche en termes de tri des variables selon leur importance, des tests préliminaires ont été effectués sur les bases de données suivantes (Blake et Merz, 1998) :

- Wine : Cette base contient les résultats d'une analyse chimique des vins produits dans la même région en Italie, mais provenant de trois cultivateurs différents (trois classes à prédire). Elle est constituée de 178 données caractérisées par 13 attributs continus.
- Letters : Cette base est constituée de 20000 données caractérisées par 16 attributs et 26 classe à prédire.
- Iris : Cette base est constituée de 150 données caractérisées par 4 attributs continus et trois classe à prédire.

4.2 Algorithme utilisé pour comparer les mesures d'importance

Une bonne mesure d'importance doit permettre de trier les variables en fonction de leur importance. Les moins bonnes de ces variables ne contiennent pas, ou peu d'information utile à la formation des clusters. Le résultat d'un clustering sur le jeu de données privé de cette variable, et donc sa qualité, devrait rester sensiblement identique, ou même être légèrement meilleur (moins de bruit). Inversement, le retrait d'une variable importante, priverait l'algo-

rithme d'une information importante pour former les clusters produisant alors un clustering de moins bonne qualité.

On définit alors un algorithme simple qui nous permet de recueillir les informations pour comparer les différentes mesures d'importance :

1. exécuter l'algorithme de clustering afin d'obtenir un premier partitionnement.
2. trier les variables selon leur importance, à l'aide de la méthode de tri que l'on souhaite tester.
3. exécuter l'algorithme de clustering afin d'obtenir un nouveau partitionnement.
4. estimer la qualité de ce partitionnement à l'aide des critères EVA et AUC.
5. retirer du jeu de donnée la variable la moins importante, d'après le tri effectué en 2.
6. réitérer à partir de l'étape 3, jusqu'à un critère d'arrêt (par exemple, toutes les variables ont été retirées).

On peut alors tracer la courbe des valeurs d'EVA (respectivement AUC (Fawcett, 2004)) en fonction du nombre de variables. L'examen des résultats peut alors être fait visuellement en observant l'évolution de la courbe des valeurs d'EVA (respectivement AUC) et/ou en calculant l'aire sous la courbe des valeurs d'EVA (respectivement AUC) (ALC = Area Under Learning Curve (Salperwyck et Lemaire, 2011)). Plus l'ALC est élevée plus la méthode de tri est de bonne qualité.

4.3 Les méthodes de tri implémentées

Nous avons listé dans la section 3.1 plusieurs indices permettant de trier les variables en fonction de leur importance dans un clustering. Pour des raisons de temps et de coût d'implémentation, à ce jour deux indices ont été implémentés à savoir Davies-Bouldin (BD) (Davies et Bouldin, 1979) et SD (Halkidi et al., 2000). Dans cette section ces deux indices seront comparés à notre approche présentée dans la section 3.2.

4.4 Résultats

Level	V3	V8	V5	V4	V2	V9	V11	V1	V6	V13	V10	V12	V7
Indice-DB	V4	V2	V13	V11	V3	V1	V10	V8	V9	V5	V12	V6	V7
Indice-SD	V4	V5	V3	V2	V13	V8	V11	V9	V1	V6	V10	V7	V12

TAB. 2 – *Tri des variables (de la moins importante à la plus importante) à l'aide des trois méthodes*

Le tableau 2 présente à titre illustratif l'ordonnancement des variables en fonction de leur importance dans le clustering obtenu, pour la base Wine, à l'aide des trois méthodes de tri implémentées. A partir de nos prétraitement [2], les deux dernières méthodes (Davies-Bouldin et SD) calculent pour chacune de ces variables trois valeurs de contribution conditionnellement à la classe à prédire. Cela veut dire qu'une seule variable peut avoir une forte contribution à la

2. Dans le cas du jeu de données wine (cas de 3 classe à prédire C_1, C_2 et C_3), chaque variable est prétraitée de la manière suivante : $log(P(X_i = x|C_1), log(P(X_i = x|C_2), log(P(X_i = x|C_3)$ avec $i \in [\![1, 13]\!]$.

construction des clusters conditionnellement à une classe à prédire et en même temps une faible contribution conditionnellement à une autre classe. Dans ce cas, on définit une contribution d'une variable comme étant la somme des trois valeurs [3]. La méthode proposée (level), est une méthode capable d'utiliser la représentation issue de prétraitement et le tableau des effectifs pour fournir une valeur de contribution par variable.

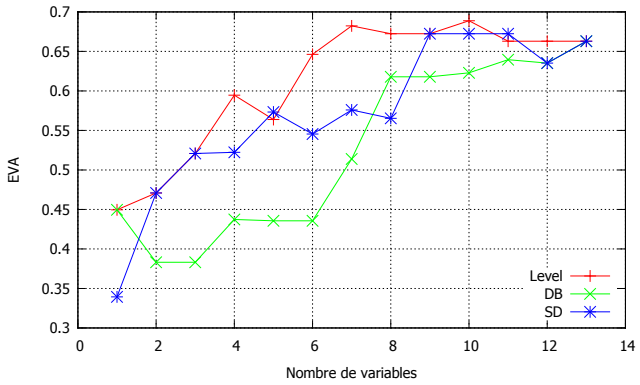


FIG. 2 – *Evolution du critère EVA pour les trois méthodes (K=3)*

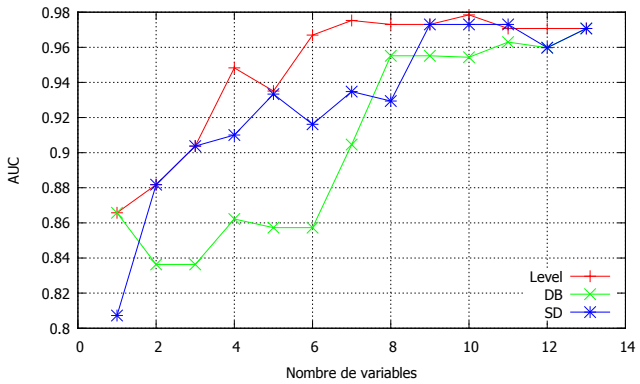


FIG. 3 – *Evolution du critère AUC pour les trois méthodes (K=3)*

La figure 2 (respectivement la figure 3) présente les trois courbes d'EVA (respectivement AUC [4]) à titre illustratif sur la base Wine en fonction de la méthode utilisée.

Le tableau 3 présente quand à lui les valeurs d'ALC pour EVA, l'AUC et l'ACC (le taux de bonne classification) selon le critère utilisé.

3. Une variable considérée comme moins contributrice pour un clustering, doit être retirée entièrement du jeu de données

4. Le critère AUC est donné par la formule suivante : $\sum_{j\in[\![1,J]\!]} P(C_j)AUC(C_j)$ avec J est le nombre de classe. Il permet de mesurer la qualité de la classification en traitant chaque cluster individuellement. Notons que la classe prédite d'un cluster est définit comme étant la classe majoritaire de celui-ci.

L'évolution du critère EVA (respectivement AUC) à l'aide de la méthode proposée est meilleure vis-à-vis de l'évolution des deux autres critères à mesure que l'on retire les variables jugées les moins contributrices pour le clustering obtenu.

		DB	SD	level
Wine	ALC(EVA)	0,5257	0,5714	0,6116
	ALC(AUC)	0,9060	0,9281	0,9472
	ALC(ACC)	0,8574	0,8863	0,9123
Letters	ALC(EVA)	0,3628	0,2871	0,3749
	ALC(AUC)	0,8930	0,8558	0,8952
	ALC(ACC)	0,3475	0,2813	0,3555
Iris	ALC(EVA)	0,6304	0,4571	0,6304
	ALC(AUC)	0,9675	0,9078	0,9675
	ALC(ACC)	0,9350	0,8267	0,9350

TAB. 3 – *Les valeurs d'ALC pour les trois méthodes selon le critère utilisé.*

On remarque également qu'il possible de trouver un nombre restreint de variables produisant la même valeur d'EVA que l'ensemble complet des variables de départ. Par exemple sur la base de données Wine, et à l'aide de la méthode proposée, on aurait pu déterminer un jeu de 7 variables qui auraient produit un clustering supervisé presque de même qualité que celui obtenu à l'aide de 13 variables.

5 Conclusion

Cette contribution a présenté une nouvelle méthode de tri des variables en cours d'élaboration dans notre contexte industriel particulier. Cette méthode trie les variables en fonction de leur importance à la fin de la convergence de notre clustering qui est "supervisé" en partie. Les résultats préliminaires qui ont été obtenus sont encourageants et semblent montrer l'intérêt de la méthode. Néanmoins ces résultats devront être confirmés sur d'avantage de base de données et comparés à un jeu de critère de qualité de la littérature de plus grande taille.

Références

Arthur, D. et S. Vassilvitskii (2007). K-means++ : The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pp. 1027–1035.

Benzécri, J. P. (1983). Analyse de l'inertie intraclasse par l'analyse d'un tableau de correspondance. pp. 351 – 358.

Blake, C. L. et C. J. Merz (1998). Uci repository of machine learning databases. last visited : 01/12/2013, http ://archive.ics.uci.edu/ml/.

Boullé, M. (2004). A Bayesian approach for supervised discretization. In Zanasi, Ebecken, et Brebbia (Eds.), *Data Mining V*, pp. 199–208. WIT Press.

Boullé, M. (2005). A grouping method for categorical attributes having very large number of values. In P. Perner et A. Imiya (Eds.), *Proceedings of the Fourth International Conference*

on Machine Learning and Data Mining in Pattern Recognition, Volume 3587 of *LNAI*, pp. 228–242. Springer verlag.

Celeux, G., E. Diday, G. Govaert, Y. Lechevallier, et H. Ralambondrainy (1989). *Classification automatique des données*. Dunod.

Davies, D. L. et D. W. Bouldin (1979). A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI-1*(2), 224–227.

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics 4*(1), 95–104.

Fawcett, T. (2004). Roc graphs : Notes and practical considerations for researchers. *Machine learning 31*(7), 1–38.

Ferrandiz, S. et M. Boullé (2010). Bayesian instance selection for the nearest neighbor rule. *Machine Learning 81*(3), 229–256.

Halkidi, M. et M. Vazirgiannis (2001). Clustering validity assessment : finding the optimal partitioning of a data set. In *Proceedings IEEE International Conference on ICDM 2001*, pp. 187–194.

Halkidi, M., M. Vazirgiannis, et Y. Batistakis (2000). Quality scheme assessment in the clustering process. In D. A. Zighed, J. Komorowski, et J. ?ytkow (Eds.), *Principles of Data Mining and Knowledge Discovery*, Volume 1910 of *Lecture Notes in Computer Science*, pp. 265–276. Springer Berlin Heidelberg.

Jajuga, K. (1987). A clustering method based on the l1-norm. *Computational Statistics & Data Analysis 5*(4), 357–371.

Kashima, H., J. Hu, B. Ray, et M. Singh (2008). K-means clustering of proportional data using l1 distance. In *19th International Conference on ICPR*.

Lemaire, V., F. Clérot, et N. Creff (2012). K-means clustering on a classifier-induced representation space : application to customer contact personalization. In *Annals of Information Systems, Springer, Special Issue on Real-World Data Mining Applications*.

Meila, M. et D. Heckerman (1998). An experimental comparison of several clustering and initialization methods. *Machine Learning*.

Rousseeuw, P. J. (1987). Silhouettes : A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics 20*(0), 53 – 65.

Salperwyck, C. et V. Lemaire (2011). Learning with few examples : An empirical study on leading classifiers. In *IJCNN*, pp. 1010–1019.

Summary

This article presents a preliminary study made in an industrial context. We have a typology of customers that the marketing service want to contact. This typology is a segmentation of customers into groups, whose profils will be used to propose differentied marketing campaigns. The constitution of groups is realised by using a clustering technique which does not currently allow the importance of the variables. This article proposes to solve this problem by using a methodology which gives in our industrial context the importance of variables. This method will be compared with some others methods from the literature.

Bibliographie

[1] A. Asuncion, D. N. UCI machine learning repository, 2007.

[2] Aggarwal, C. C., and Reddy, C. K. *DATA CLUSTERING Algorithms and Applications*. Data Mining and Knowledge Discovery Series, 2013.

[3] Agrawal, R., and Srikant, R. Fast algorithms for mining association rules. In *Proc. of 20th Intl. Conf. on VLDB* (1994), pp. 487–499.

[4] Aguilar, J. S., Ruiz, R., Riquelme, J. C., and Giráldez, R. Snn : A supervised clustering algorithm. In *Engineering of Intelligent Systems*. Springer, 2001, pp. 207–216.

[5] Al-Daoud, M. B. A new algorithm for cluster initialization. In *WEC'05 : The Second World Enformatika Conference* (2005).

[6] Al-Harbi, S. H., and Rayward-Smith, V. J. Adapting k-means for supervised clustering. *Applied Intelligence 24*, 3 (2006), 219–226.

[7] Al Hasan, M., Chaoji, V., Salem, S., and Zaki, M. J. Robust partitional clustering by outlier and density insensitive seeding. *Pattern Recognition Letters 30*, 11 (2009), 994–1002.

[8] Alaoui Ismaili, O., Lemaire, V., and Cornuéjols, A. Une méthode basée sur des effectifs pour calculer la contribution des variables à un clustering. In *Atelier CluCo de la conférence Extraction et Gestion des Connaissances (EGC 2014), Rennes* (2014).

[9] Alaoui Ismaili, O., Lemaire, V., and Cornuéjols, A. Classification à base de clustering ou comment décrire et prédire simultanément. In *Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJCIA)* (2015), pp. 7–12.

[10] Alaoui Ismaili, O., Lemaire, V., and Cornuéjols, A. Supervised pre-processings are useful for supervised clustering. *Springer Series Studies in Classification, Data Analysis, and Knowledge Organization* (2015).

[11] Alaoui Ismaili, O., Lemaire, V., and Cornuéjols, A. Une méthode supervisée d'initialisation des centres pour les k-moyennes. *Congrès de la Société Française de Classification (SFC)* (2015).

[12] Alaoui Ismaili, O., Lemaire, V., and Cornuéjols, A. Clustering prédictif : décrire, prédire et interpréter simultanément. In *à venir sur invitation suite à la conférence RJCIA, in Revue d'intelligence Artificielle (RIA)* (2016).

[13] Alaoui Ismaili, O., Lemaire, V., and Cornuéjols, A. Evaluation of predictive clustering quality. In *in MBC2, on Model Based clustering and classification (MBC2,2016)* (2016).

[14] Andrews, R., Diederich, J., and Tickle, A. B. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowl.-Based Syst. 8*, 6 (1995), 373–389.

[15] Arthur, D., and Vassilvitskii, S. K-means++ : The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (2007), Society for Industrial and Applied Mathematics, pp. 1027–1035.

[16] Bader, S., Hölldobler, S., and Mayer-Eichberger, V. Extracting propositional rules from feed-forward neural networks-a new decompositional approach. In *NeSy* (2007).

[17] Ball, G. H., and Hall, D. J. A clustering technique for summarizing multivariate data. *Behavioral science 12*, 2 (1967), 153–155.

[18] Basu, M., and Ho, T. K. *Data complexity in pattern recognition*. Springer Science & Business Media, 2006.

[19] Basu, S., Banerjee, A., and Mooney, R. J. Semi-supervised clustering by seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning* (San Francisco, CA, USA, 2002), ICML '02, Morgan Kaufmann Publishers Inc., pp. 27–34.

[20] Benki, A. Méthodes efficaces de capture de front de pareto en conception mécanique multicritère. In *applications industrielles. Ph. D. thesis.* (2014).

[21] Bertier, P., Bouroche, J.-M., and Morlat, G. Analyse des données multidimensionnelles.

[22] Boley, D. Principal direction divisive partitioning. *Data mining and knowledge discovery 2*, 4 (1998), 325–344.

[23] Boullé, M. MODL : a Bayes optimal discretization method for continuous attributes. *Machine Learning 65*, 1 (2006), 131–165.

[24] Bradley, P. S., and Fayyad, U. M. Refining initial points for k-means clustering. In *ICML* (1998), vol. 98, Citeseer, pp. 91–99.

[25] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. *Classification and regression trees*. CRC press, 1984.

[26] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. Lof : identifying density-based local outliers. In *ACM sigmod record* (2000), vol. 29, ACM, pp. 93–104.

[27] Briesemeister, S., Rahnenführer, J., and Kohlbacher, O. Going from where to why—interpretable prediction of protein subcellular localization. *Bioinformatics 26*, 9 (2010), 1232–1238.

[28] Celebi, M. E., and Kingravi, H. A. Deterministic initialization of the k-means algorithm using hierarchical clustering. *International Journal of Pattern Recognition and Artificial Intelligence 26*, 07 (2012), 1250018.

[29] Celebi, M. E., and Kingravi, H. A. Linear, deterministic, and order-invariant initialization methods for the k-means clustering algorithm. *CoRR abs/1409.3854* (2014).

[30] Celebi, M. E., Kingravi, H. A., and Vela, P. A. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Syst. Appl. 40*, 1 (2013), 200–210.

[31] Celebi, M. E., Kingravi, H. A., and Vela, P. A. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications 40*, 1 (2013), 200–210.

[32] Cevikalp, H., Larlus, D., and Jurie, F. A supervised clustering algorithm for the initialization of rbf neural network classifiers. In *Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th* (2007), IEEE, pp. 1–4.

[33] Chen, F. Learning accurate and understandable rules from svm classifiers. *Master's thesis, Simon Fraser University* (2004).

[34] Clark, P., and Niblett, T. The cn2 induction algorithm. *Machine learning 3*, 4 (1989), 261–283.

[35] Cornuéjols, A., and Miclet, L. *Apprentissage artificiel : Concepts et algorithmes.* Eyrolles, June 2010.

[36] Craven, M., and Shavlik, J. Rule extraction : Where do we go from here. *University of Wisconsin Machine Learning Research Group working Paper* (1999), 99–1.

[37] Cunningham, P., and Delany, S. J. k-nearest neighbour classifiers. *Mult Classif Syst* (2007), 1–17.

[38] Davies, D. L., and Bouldin, D. W. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell. 1*, 2 (1979), 224–227.

[39] de Carvalho, F. d. A., Saporta, G., and Queiroz, D. N. A clusterwise center and range regression model for interval-valued data. In *Proceedings of COMPSTAT'2010.* Springer, 2010, pp. 461–468.

[40] De Roover, K., Ceulemans, E., Timmerman, M. E., Vansteelandt, K., Stouten, J., and Onghena, P. Clusterwise simultaneous component analysis for analyzing structural differences in multivariate multiblock data. *Psychological methods 17*, 1 (2012), 100.

[41] Demšar, J. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research 7* (2006), 1–30.

[42] DeSarbo, W. S., and Cron, W. L. A maximum likelihood methodology for clusterwise linear regression. *Journal of classification 5*, 2 (1988), 249–282.

[43] Dimitrovski, I., Kocev, D., Loskovska, S., and Dzeroski, S. Fast and efficient visual codebook construction for multi-label annotation using predictive clustering trees. *Pattern Recognition Letters 38* (2014), 38–45.

[44] Ding, C., and He, X. Cluster merging and splitting in hierarchical clustering algorithms. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on* (2002), IEEE, pp. 139–146.

[45] Dom, B. E., and Dom, B. E. An information-theoretic external cluster-validity measure. Tech. rep., Research Report RJ 10219, IBM, 2001.

[46] Eick, C. F., Zeidat, N., and Zhao, Z. Supervised clustering-algorithms and benefits. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on* (2004), IEEE, pp. 774–776.

[47] Féraud, R., and Clérot, F. A methodology to explain neural network classification. *Neural Networks 15*, 2 (2002), 237–246.

[48] Ferrandiz, S., and Boullé, M. Bayesian instance selection for the nearest neighbor rule. *Machine Learning 81*, 3 (2010), 229–256.

[49] Forgy, E. W. Cluster analysis of multivariate data : efficiency versus interpretability of classifications. *Biometrics 21* (1965), 768–769.

[50] Freitas, A. A. Comprehensible classification models : A position paper. *SIGKDD Explor. Newsl. 15*, 1 (Mar. 2014), 1–10.

[51] Fung, G., Sandilya, S., and Rao, R. B. Rule extraction from linear support vector machines. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (2005), ACM, pp. 32–40.

[52] Gonzalez, T. F. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science 38* (1985), 293–306.

[53] Guyon, I., Luxburg, U. V., and Williamson, R. C. Clustering : Science or art. In *NIPS 2009 Workshop on Clustering Theory* (2009).

[54] Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. Gene selection for cancer classification using support vector machines. *Machine learning 46*, 1-3 (2002), 389–422.

[55] Halkidi, M., Vazirgiannis, M., and Batistakis, Y. Quality scheme assessment in the clustering process. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery* (London, UK, UK, 2000), PKDD '00, Springer-Verlag, pp. 265–276.

[56] Hartigan, J. A., and Wong, M. A. Algorithm as 136 : A k-means clustering algorithm. *Applied statistics* (1979), 100–108.

[57] Hubert, L., and Arabie, P. Comparing partitions. *Journal of Classification 2*, 1 (Dec. 1985), 193–218.

[58] Huysmans, J., Baesens, B., and Vanthienen, J. Iter : an algorithm for predictive regression rule extraction. In *Data Warehousing and Knowledge Discovery*. Springer, 2006, pp. 270–279.

[59] Ismaili, O. A., Lemaire, V., and Cornuéjols, A. A supervised methodology to measure the variables contribution to a clustering. In *Neural Information Processing - 21st International Conference, ICONIP 2014, Kuching, Malaysia, November 3-6, 2014. Proceedings, Part I* (2014), pp. 159–166.

[60] Ismaili, O. A., Lemaire, V., and Cornuéjols, A. Une méthode supervisée pour initialiser les centres des k-moyennes. In *16ème Journées Francophones Extraction et Gestion des Connaissances, EGC 2016, 18-22 Janvier 2016, Reims, France* (2016), pp. 147–152.

[61] Jain, A. K. Data clustering : 50 years beyond k-means. *Pattern Recogn. Lett. 31*, 8 (June 2010), 651–666.

[62] Jain, A. K., Murty, M. N., and Flynn, P. J. Data clustering : A review. *ACM Comput. Surv. 31*, 3 (Sept. 1999), 264–323.

[63] Johansson, U., Konig, R., and Niklasson, L. Automatically balancing accuracy and comprehensibility in predictive modeling. In *Information Fusion, 2005 8th International Conference on* (2005), vol. 2, IEEE, pp. 7–pp.

[64] Katsavounidis, I., Jay Kuo, C.-C., and Zhang, Z. A new initialization technique for generalized lloyd iteration. *Signal Processing Letters, IEEE 1*, 10 (1994), 144–146.

[65] Kaufman, L., and Rousseeuw, P. J. *Finding groups in data : an introduction to cluster analysis*, vol. 344. John Wiley & Sons, 2009.

[66] Kodinariya, T. M., and Makwana, P. R. Survey on exiting method for selecting initial centroids in k-means clustering. In *International Journal of Engineering Development and Research* (2014), vol. 2, IJEDR.

[67] Kononenko, I., Bratko, I., and Roskar, E. Experiments in automatic learning of medical diagnostic rules. Tech. rep., E. Kardelj University, Faculty of Electrical Engineering, Ljubljana, Yugoslavia, 1984.

[68] Kotsiantis, S. B. Supervised machine learning : A review of classification techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering : Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies* (Amsterdam, The Netherlands, The Netherlands, 2007), IOS Press, pp. 3–24.

[69] Lemaire, V., and Ismaili, O. A. Un outil pour la classification à base de clustering pour décrire et prédire simultanément. In *Atelier Clustering and Co-clustering (CluCo), EGC, Reims* (2016).

[70] Lemaire, V., Ismaili, O. A., and Cornuéjols, A. An initialization scheme for supervized k-means. In *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17, 2015* (2015), pp. 1–8.

[71] Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. Understanding of internal clustering validation measures. In *Proceedings of the 2010 IEEE International Conference on Data Mining* (Washington, DC, USA, 2010), ICDM '10, IEEE Computer Society, pp. 911–916.

[72] MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability - Vol. 1* (1967), L. M. Le Cam and J. Neyman, Eds., University of California Press, Berkeley, CA, USA, pp. 281–297.

[73] Manning, C. D., Raghavan, P., Schütze, H., et al. *Introduction to information retrieval*, vol. 1. Cambridge university press Cambridge, 2008.

[74] Markowska-Kaczmar, U., and Chumieja, M. Discovering the mysteries of neural networks. *International Journal of Hybrid Intelligent Systems 1*, 3 (2004), 153–163.

[75] Martens, D., Baesens, B., Van Gestel, T., and Vanthienen, J. Comprehensible credit scoring models using rule extraction from support vector machines. *European journal of operational research 183*, 3 (2007), 1466–1476.

[76] Meilă, M. Comparing clusterings by the variation of information. In *Learning theory and kernel machines*. Springer, 2003, pp. 173–187.

[77] Milligan, G., and Cooper, M. A study of standardization of variables in cluster analysis. *Journal of Classification 5*, 2 (1988), 181–204.

[78] Nazeer, K. A., and Sebastian, M. Improving the accuracy and efficiency of the k-means clustering algorithm. In *Proceedings of the World Congress on Engineering* (2009), vol. 1, pp. 1–3.

[79] Núñez, H., Angulo, C., and Català, A. Support vector machines with symbolic interpretation. In *SBRN* (2002), IEEE Computer Society, pp. 142–149.

[80] Nguyen, D., and Le, M. Improving the interpretability of support vector machines-based fuzzy rules. *CoRR abs/1408.5246* (2014).

[81] núria macià antolínez. Data complexity in supervised learning : A far-reaching implication. In *PhD Thesis* (2012).

[82] Ocegueda-Hernandez, F., and Vilalta, R. An empirical study of the suitability of class decomposition for linear models : When does it work well ? In *SDM* (2013), SIAM, pp. 432–440.

[83] Oh, I.-S., Lee, J.-S., and Moon, B.-R. Hybrid genetic algorithms for feature selection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 26*, 11 (2004), 1424–1437.

[84] Peng, H., Long, F., and Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 27*, 8 (2005), 1226–1238.

[85] Peralta, B., Espinace, P., and Soto, A. Enhancing k-means using class labels. *Intell. Data Anal. 17*, 6 (Nov. 2013), 1023–1039.

[86] Preda, C., and Saporta, G. Clusterwise pls regression on a stochastic process. *Computational Statistics & Data Analysis 49*, 1 (2005), 99–108.

[87] Quinlan, J. R. *C4.5 : programs for machine learning*, vol. 1. Morgan kaufmann, 1993.

[88] Rencher, A. C. *Methods of Multivariate Analysis*. Wiley, 2003.

[89] R.L.Burden, and J.D.Faires. *Numerical Analysis*. Weber, Schmidt,Boston, MA,, 1985.

[90] Robnik-Sikonja, M., and Kononenko, I. Explaining classifications for individual instances. *Knowledge and Data Engineering, IEEE Transactions on 20*, 5 (2008), 589–600.

[91] Rousseeuw, P. Silhouettes : A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math. 20*, 1 (Nov. 1987), 53–65.

[92] Saito, K., and Nakano, R. Extracting regression rules from neural networks. *Neural Networks 15*, 10 (2002), 1279–1288.

[93] Setiono, R., Leow, W. K., and Zurada, J. M. Extraction of rules from artificial neural networks for nonlinear regression. *IEEE Transactions on Neural Networks 13*, 3 (2002), 564–577.

[94] Sindhwani, V., Rakshit, S., Deodhare, D., Erdogmus, D., Principe, J. C., and Niyogi, P. Feature selection in mlps and svms based on maximum output information. *Neural Networks, IEEE Transactions on 15*, 4 (2004), 937–948.

[95] Slonim, N., and Tishby, N. Agglomerative information bottleneck. MIT Press, pp. 617–623.

[96] Strehl, A., and Ghosh, J. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res. 3* (Mar. 2003), 583–617.

[97] Su, T., and Dy, J. G. In search of deterministic methods for initializing k-means and gaussian mixture clustering. *Intelligent Data Analysis 11*, 4 (2007), 319–338.

[98] Terry Therneau, Beth Atkinson, B. R. Recursive partitioning and regression trees. In *https ://cran.r-project.org/web/packages/rpart/rpart.pdf* (2015).

[99] Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057* (1999).

[100] Tou, J. T., and Gonzalez, R. C. Pattern recognition principles. *Pattern Recognition in Physics 1* (1974).

[101] Vilalta, R., and Rish, I. A Decomposition Of Classes Via Clustering To Explain And Improve Naive Bayes. In *Proceedings of 14th European Conference on Machine Learning (ECML/PKDD 2003)* (2003).

[102] Wu, J., Xiong, H., and Chen, J. Adapting the right measures for k-means clustering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2009), KDD '09, ACM, pp. 877–886.

[103] Xiuju, F., ChongJin, O., S, K., Gih G, H., and Liping, G. Extracting the knowledge embedded in support vector machines. In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2004), pp. 291–296.

[104] Zhang, Y., Su, H., Jia, T., and Chu, J. Rule extraction from trained support vector machines. In Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 92–95.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Agriculture, alimentation, biologie, environnement, santé (ABIES)

Titre : Clustering prédictif : Décrire et Prédire simultanément

Mots clés : Description, prédiction, interprétation, clustering, classification supervisée, clustering prédictif

Résumé : Le clustering prédictif est un nouvel aspect d'apprentissage supervisé dérivé du clustering standard. Les algorithmes appartenant à ce type de l'apprentissage cherchent à décrire et à prédire d'une manière simultanée. Il s'agit de découvrir la structure interne d'une variable cible. Puis munis de cette structure, de prédire la classe des nouvelles instances.
Pour atteindre l'objectif de la thèse qui est la recherche d'un modèle d'apprentissage "interprétable" capable de décrire et de prédire d'une manière simultanée, nous avons choisi de modifier l'algorithme des K-moyennes standard. Cette version modifiée est nommée *les K-moyennes prédictives*. Ellle contient 7 différentes étapes dont chacune peut être supervisée indépendamment des autres.
Au cours de cette thèse, nous nous intéressons à la supervision de quatre étapes, à savoir : 1) le prétraitement des données, 2) l'initialisation des centres, 3) le choix de la meilleure partition et 4) la mesure d'importance des variables.
Nos résultats expérimentaux montrent d'une part qu'avec la supervision de l'étape de prétraitement des données et de l'étape d'initialisation des centres, l'algorithme des K-moyennes prédictives parvient à avoir des performances très compétitives ou meilleures que celles obtenues par certains algorithmes de clustering prédictif.
D'autre part, ces résultats expérimentaux mettent l'accent sur la capacité de nos méthodes de prétraitement à aider l'algorithme des K-moyennes prédictives à fournir des résultats facilement interprétables par l'utilisateur.
Nous montrons enfin dans ce mémoire qu'avec l'aide du critère d'évaluation proposé dans cette thèse, l'algorithme des K-moyennes prédictives parvient à sélectionner la partition optimale qui réalise le bon compromis entre la description et la prédiction. Ceci permet à l'utilisateur de découvrir les différentes raisons qui peuvent mener à une même prédiction.

Title : Predictive clustering : Describe and Predict simultaneously

Keywords : Description, prediction, interpretation, clustering, supervised learning, predictif clustering

Abstract : Predictive clustering is a new supervised learning framework derived from traditional clustering. This new framework allows to describe and to predict simultaneously. Compared to a classical supervised learning, predictive clsutering algorithms seek to discover the internal structure of the target class in order to use it for predicting the class of new instances.
The purpose of this thesis is to look for an interpretable model of predictive clustering. To acheive this objective, we choose to modified traditional K-means algorithm. This new modified version is called *predictive K-means*. It contains 7 differents steps, each of which can be supervised seperatly from the others. In this thesis, we only deal four steps : 1) data preprocessing, 2) initialization of centers, 3) selecting of the best partition, and 4) importance of features.
Our experimental results show that the use of just two supervised steps (data preprocessing and initialization of centers), allow the K-means algorithm to acheive competitive performances with some others predictive clustering algorithms.
These results show also that our preprocessing methods can help predictive K-means algorithm to provide results easily comprehensible by users.
We are also showing in this thesis that the use of our new measure to evaluate predictive clustering quality, helps our predictive K-means algorithm to find the optimal partition that establishes the best trade-off between description and prediction. It thus allows users to find the different reasons behind the same prediction : two differents instances could have the same predicted label.