Détection et localisation d'anomalies dans des données hétérogènes en utilisant des modèles graphiques non orientés mixtes

Romain Laby

▶ **To cite this version:**

Romain Laby. Détection et localisation d'anomalies dans des données hétérogènes en utilisant des modèles graphiques non orientés mixtes. Machine Learning [stat.ML]. Télécom ParisTech, 2017. Français. NNT : 2017ENST0026 . tel-03219690

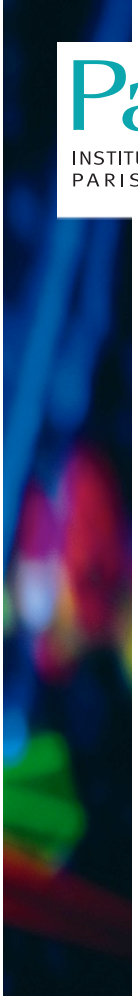
HAL Id: tel-03219690

https://pastel.hal.science/tel-03219690

Submitted on 6 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2017-ENST-0026

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Mathématiques Appliquées »

présentée et soutenue publiquement par

Romain LABY

le 24 mai 2017

Détection et localisation d'anomalies par utilisation de modèles graphiques mixtes

Directeur de thèse : **François ROUEFF**
Co-encadrement de la thèse : **Alexandre GRAMFORT**

Jury

M. Wojciech PIECZYNSKI, Professeur, TELECOM SudParis	Président du jury
M. Yves GRANDVALET, Chercheur CNRS, UTC	Examinateur
M. Guillaume OBOZINSKI, Maître de conférence, ENPC Paristech	Examinateur
Mme Marianne CLAUSEL, Maître de conférence, Université de Grenoble	Examinatrice
M. Christophe AMBROISE, Professeur, Université d'Évry Val d'Essonne	Rapporteur
Mme Sophie ACHARD, Directrice de recherche CNRS, CNRS	Rapporteuse
M. Cyrille ENDERLI, Ingénieur, Thales Systèmes Aéroportés	Examinateur

TELECOM ParisTech
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

T H È S E

Contents

List of Figures

List of Algorithms

Preface

I spend three splendid years working on my PhD in Paris, and now that this journey has ended, I remember all the people who accompanied me, who taught quite everything I learned, who supported me, or simply were there during the good and the hard time, and I would like to sincerely thank every one of them. Folks, I owe you a lot.

I would first like to express my sincerest acknowledgements to my advisors, François Roueff and Alexandre Gramfort. François has always impressed me with his ideas, his intuitions, his precision. I was amazed by his way of considering problems and finding simple and elegant solutions. Alexandre was my mentor in many ways during these three years. By his side, I learned how to transform ideas and drafts into accurate algorithms and efficient applications. Under their watch and dedication, I stayed all along excited about my research (even though I often understood their ideas only days after they exposed them to me), and they made me grow up in many aspects. They picked me as a young boy at ease with baloney, and I returned, I hope, stronger in science, and wiser about myself.

This thesis would not have an exciting industrial adventure if not for Alain Larroque and Cyrille Enderli. Alain has always overwhelmed me by constantly having a big vision in mind, from the groundwork he laid, to the future applications of the project he had (when I think that my algorithms might flight one day...). Cyrille dedicated a lot of his time to support me and the doctoral students in Thales. I enjoyed a lot discussing with him about my research, even when we discovered that things were not so simple as I thought they were ! They both fully trusted me and backed me all the time, and I give them my deepest thanks.

I want to thank all the administrative people from Thales and Telecom that I've met, who helped me during my PhD and made me kind of feel home: Colette, Dominique, Liliane, Laurence, Florence and Marianna.

These years would not have been so great if I wasn't supported by my friends. The paper planes we designed (best paper planes ever made!), the Nerf Guns and the traps, the Zerg rushes and the zombies, you all made it an incredible story. Raphaël, Mathieu, Raphaël, Clément, Véronique, Florian, Pauline, Michaël, Mathilde, Damien, Guillaume, Anne, Antoine, Marie-Agathe, Michal, Ludovic, Hugo, Philippe, Guillaume, Costin, Gaëlle, Juliette, Claire, Cyrille, Théo, Camille, Pascal, thank you for the marvellous time we spent together. I would like to give

a special thank to Raphaël, Michal and Florian, with who I discussed a lot and who spent time enlightening many misunderstanding and answering many questions I had. I also need to mention all the people from the production site near Bordeaux I have worked with: Théo, Sylvain, Eric, Alain, Christophe, Colette, José, Sébastien. That time when we run aground in the middle of the bay of Arcachon is part of the best memories I have!

I am indebted to all my family, especially my mum Christiane, my dad Christian, my sister Sophiane, who always been there for me, who provided me with everything I needed and supported me every day. I would like to thank my grandfather Roman and my grandmother Irène, who did not make it until the end of my thesis but might still watch it from above, and all the others, especially Thierry and Jean-Luc.

I have also a special though for my former mathematics teacher, Alain Juhel. I owe him a lot of the passion and excitement I feel today with research and science. And I've also never forgotten to drawn krobars !

Chapter 1

Résumé en Français

1.1 Introduction

Dans cette section, nous allons introduire le problème de détection d'anomalie qui a motivé cette étude. Notre travail s'inscrit dans une étude plus vaste réalisée chez Thales Systèmes Aeroportés, qui a débuté en 2010, quand a grandi l'idée d'utiliser des techniques d'apprentissage automatique pour compléter le système de détection de panne qui équipe les radars produits à Thales. Nous présenterons le problème industriel, l'équipement concerné, les données qu'ils produisent lors de leur utilisation, quel est le système de détection actuel et les motivations industrielles qui ont mené à notre étude.

Thales est un groupe mondial spécialisé dans l'aéronautique, l'espace, le transport terrestre, la sécurité et la défense. Thales Systèmes Aeroportés, appartenant à la division aéronautique, développe des systèmes qui répondent à de nombreux besoins opérationnels: systèmes embarqués, sous-systèmes, systèmes ou services complets, pour les clients militaires et civils. Dans le cadre de son expertise, cette entreprise française surveille de nombreux programmes militaires et travaille comme sous-traitant pour des programmes d'autres entreprises, comme Dassault Mirage et Dassault Rafale, deux avions de chasse français produits par Dassault Aviation. Plus précisément pour ces deux avions, Thales développe et produit certaines de ses composants électroniques, parmi lesquelles leur radar de combat, les systèmes de défense intégrés, les systèmes de gestion de vol, le contrôle de tir, les interfaces pour pilotes, les pods, les capteurs, etc.

Cette thèse répond à un besoin spécifique concernant le radar RBE2 (Radar à Balayage Électronique 2 plans), qui est le radar de combat équipant le Rafale. A de nombreux titres, il est plus performant et complexe que les radars de générations antérieures.

1.1.1 La maintenance intégrée

Un radar de combat est composé de nombreux sous-systèmes : l'antenne, le module hyperfréquence, des blocs de traitement du signal, un système de refroidissement, des interfaces avec l'avion, entre autres. Tous ces composants doivent fonctionner de manière optimale dans de nombreuses situations extrêmes, par exemple lors de fortes vibrations, des accélérations brutales, où en présence de forte humidité, fortes pressions ou températures. Ces conditions de fonctionnement peuvent causer des dégâts ou un vieillissement précoce. Pour contrôler leur état de santé, chaque radar est équipé d'un système de maintenance, appelé maintenance intégrée, qui a pour objectifs :

- Évaluer l'état de fonctionnement du radar et informer les autres systèmes de l'avion,
- Détecter et localiser les éléments en panne,
- Produire des rapports exploitables par des équipes expertes pour des investigations futures.

1.2 Apprentissage d'un modèle graphique

1.2.1 Présentation du modèle mixte pair-à-pair

Dans ce chapitre, nous introduisons les modèles graphiques mixtes et comment les apprendre à partir de données. Les données que nous examinons sont des échantillons de variables hétérogènes : certaines variables sont quantitatives et peuvent représenter des gains, des phases ou des températures, et d'autres variables sont catégorielles et peuvent représenter des états ou des modes de fonctionnement. Dans les sections suivantes, on désignera par X une instance des variables x, avec $x = (x_{\mathcal{C}}; x_{\mathcal{Q}})$ où $x_{\mathcal{C}} = (x_i; i \in \mathcal{C})$ sont les variables catégorielles et où $x_{\mathcal{Q}} = (x_u; u \in \mathcal{Q})$ sont les variables quantitatives. Ici $\mathcal{C}$ et $\mathcal{Q}$ représentent respectivement les indices des variables catégoriques et quantitatives de x.

Comme expliqué précédemment dans la section 1.1, les modèles graphiques non orientés pair-à-pair offrent de nombreux avantages. La restriction aux modèles pair-à-pairs est un bon compromis entre richesse des modèles disponibles et complexité d'apprentissage. Cette complexité est polynomiale et rend l'apprentissage envisageable en grande dimension.

Le modèle proposé est un modèle mélangeant un modèle Gaussien, utilisé quand les variables sont continues et à valeur dans $\mathbb{R}$, et un modèle d'Ising, utilisé quand les variables sont binaires à valeur dans $\{-1, 1\}$ ou $\{0, 1\}$. Un modèle Gaussien est paramétré par une matrice de covariance

Σ définie positive et un vecteur moyenne μ, et sa densité est donné par

$$p_{\Sigma,\mu}(x) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right].$$

Cette densité peut se réécrire en utilisant la matrice de précision $\Delta = \Sigma^{-1}$, et est donnée par

$$p_{\Delta,\mu}(x) = \frac{1}{Z_{\Delta,\mu}} \exp\left[\mu^T\Delta x - \frac{1}{2}x^T\Delta x\right], \tag{1.1}$$

, où ici $Z_{\Delta,\mu}$ est la constante de normalisation. Un modèle d'Ising est paramétré par une matrice symétrique $\Theta = \{(\theta_i)_{i=1\ldots n}, (\theta_{ij})_{i>j})\}$, et sa distribution est donnée par

$$p_\Theta(x) = \frac{1}{Z_\Theta} \exp\left[\sum_{i=1}\theta_i x_i + \sum_{i>j}\theta_{ij}x_i x_j\right], \tag{1.2}$$

où Z_Θ est la constante de normalisation. Dans ce rapport, nous utiliserons $\{0,1\}$ comme domaine pour les variables binaires x_i, ce qui implique notamment $x_i = x_i^2$. La densité d'un modèle d'Ising peut ainsi se réécrire

$$p_\Theta(x) = \frac{1}{Z_\Theta} \exp\left[\sum_{i,j=1}^{n}\theta_{ij}x_i x_j\right] = \frac{1}{Z_\Theta}\exp(x^T\Theta x). \tag{1.3}$$

Ici nous considérons uniquement des variables binaires, et non pas des variables catégorielles (qui peuvent être non binaires) . Il est possible d'utiliser un modèle de Potts (voir Potts [1953]) avec des variables catégorielles, cependant ce modèle ne peut être utilisé que si toutes ces variables ont le même domaine et si les états des différentes variables sont comparables, ce qui n'est pas le cas des variables du RBE2. Pour résoudre ce problème, on peut binariser les variables catégorielles non binaires avec le schéma classique *one-hot*, comme suggéré par [Bishop, 2006, §4.3.4] et Schmidt [2010]. Le principe est le suivant: pour $i \in \mathcal{C}$, si x_i prends des valeurs dans $1,\ldots,m_i$, on utilisera à la place le vecteur $t^{(i)} \in \{0,1\}^{m_i}$, avec $t^{(i)}_{k_0} = 1$ si $x_i = k_0$, et $t^{(i)}_k = 0$ ailleurs, pour $k \neq k_0$. Cette transformation est réalisée uniquement pour les variables catégorielles et ne va donc impacter que les paramètres Θ and Φ, dont les dimensions vont augmenter en conséquence. Dans la suite, quand on utilisera les notations X, x, $X_\mathcal{C}$ et $x_\mathcal{C}$, on supposera que les variables catégorielles non binaires ont déjà été binarisées.

On peut maintenant introduire le framework des modèles graphiques non orientés mixtes. Pour des variables hétérogènes $x = (x_\mathcal{C}, x_\mathcal{Q})$ avec $x_\mathcal{C} \in \{0,1\}^{|\mathcal{C}|}$ and $x_\mathcal{Q} \in \mathbb{R}^{|\mathcal{Q}|}$, on défini la

densité d'un modèle mixte non orienté pair-à-pair par

$$p_\Omega(x) = \frac{1}{Z_\Omega} \exp\left[x_\mathcal{C}^T \Theta x_\mathcal{C} + \mu^T x_\mathcal{Q} - \frac{1}{2} x_\mathcal{Q}^T \Delta x_\mathcal{Q} + x_\mathcal{C}^T \Phi x_\mathcal{Q}\right], \tag{1.4}$$

où $\Omega = (\Theta, \mu, \Delta, \Phi)$ contient tous les paramètres du modèle, et où Z_Ω est la fonction de partition définie par

$$Z_\Omega = \sum_{x_\mathcal{C} \in \{0,1\}^{|\mathcal{C}|}} \int_{\mathbb{R}^{|\mathcal{Q}|}} \exp\left[x_\mathcal{C}^T \Theta x_\mathcal{C} + \mu^T x_\mathcal{Q} - \frac{1}{2} x_\mathcal{Q}^T \Delta x_\mathcal{Q} + x_\mathcal{C}^T \Phi x_\mathcal{Q}\right] dx_\mathcal{Q}. \tag{1.5}$$

Ici, $\Theta = (\theta_{ij})_{i,j\in\mathcal{C}}$ est une matrice symétrique, $\mu = (\mu_i)_{i\in\mathcal{Q}} \in \mathbb{R}^\mathcal{Q}$, $\Delta = (\delta_{uv})_{u,v\in\mathcal{Q}}$ est une matrice symétrique définie positive et $\Phi = (\phi_{iu})_{i,u\in\mathcal{C}\times\mathcal{Q}}$ est une matrice quelconque. Pour simplifier les notations, nous avons confondu les indices des variables $\mathcal{C}$ and $\mathcal{Q}$ avec les indices des matrices correspondantes.

Le modèle (1.4) a d'intéressantes propriétés. On dénote par $x_\mathcal{C}$ et $x_\mathcal{Q}$ les variables binaires et quantitatives, respectivement, avec $x = (x_\mathcal{C}, xq)$ où $x_\mathcal{C} \in \{0,1\}^{|\mathcal{C}|}$ and $x_\mathcal{Q} \in \mathbb{R}^{|\mathcal{Q}|}$. Les quatre propriétés suivantes sont valables pour n'importe quel modèle mixte pair-à-pair (1.4) paramétré par $\Omega = (\Theta, \mu, \Delta, \Phi)$:

i. Etant donné $x_\mathcal{C}$, la distribution conditionnelle de $x_\mathcal{Q}$ est gaussienne de moyenne $\Delta^{-1}\left(\mu + \Phi^T x_\mathcal{C}\right)$ et de matrice de covariance Δ^{-1}.

ii. Etant donné $x_\mathcal{Q}$, la distribution conditionnelle de $x_\mathcal{C}$ est un modèle d'Ising de paramètres $\Theta + \text{Diag}\left(\Phi x_\mathcal{Q}\right)$.

iii. La distribution marginale de $x_\mathcal{C}$ est un modèle d'Ising de paramètres $\Theta + \Phi\Delta^{-1}\Phi^T/2 + \text{Diag}\left(\Phi\Delta^{-1}\mu\right)$.

iv. La distribution marginale de $x_\mathcal{Q}$ est un mélange de modèles Gaussien, sauf quand $\Phi = 0$, dans quel cas elle est gaussienne de moyenne $\Delta^{-1}\mu$ et de matrice de covariance Δ^{-1}.

Deux de ces propriétés sont illustrées sur la Figure 1.1, où on illustres quelques simulations de la densité mixte (1.4) dans le cas $\Phi = 0$ (à gauche) et dans le cas $\Phi \neq 0$ (à droite). Le processus d'échantillonnage sera décris ultérieurement par l'algorithme **??**. La propriété 1.0.i, indiquant que les variables quantitatives ont conditionnellement une distribution gaussienne, est illustrée sur les deux figure, où les échantillons de même couleur proviennent de la même distribution. La propriété 3.1.iv, indiquant que la densité marginale de $x_\mathcal{Q}$ est gaussienne seulement si $\Phi = 0$, est illustrée sur les deux figures: à gauche, où $\Phi = 0$, les échantillons quantitatifs ont effectivement une distribution gaussienne, tandis qu'à droite, où $\Phi \neq 0$, les échantillons n'ont pas une densité gaussienne multivariée mais un mélange de distributions gaussiennes.

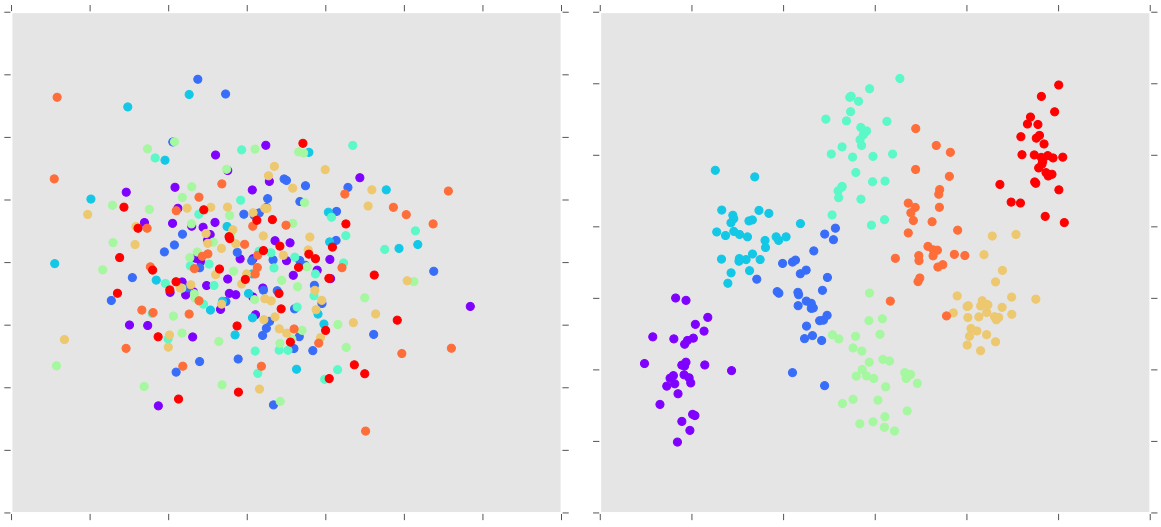


Figure 1.1: Illustrations d'échantillons i.i.d. de la densité mixte (1.4) avec 2 variables quantitative et 3 variables binaires. Sur les deux figures, les variables binaires sont représentées par $2^3 = 8$ couleurs, et les deux variables quantitatives sont affichées dans le plan. La figure de gauche illustre le cas $\Phi = 0$, i.e., quand $x_{\mathcal{Q}}$ et $x_{\mathcal{C}}$ sont indépendants, et la figure de droite illustre le cas $\Phi \neq 0$, i.e., quand $x_{\mathcal{Q}}$ et $x_{\mathcal{C}}$ sont dépendants.

1.2.2 Apprentissage d'un modèle

Nous avons développé deux algorithmes pour apprendre la structure et les paramètres d'un modèle à partir de données : optimisation de la vraisemblance pénalisée en utilisant une version stochastique du gradient proximal, et optimisation de la pseudo vraisemblance pénalisée en utilisant la version classique du gradient proximal.

L'apprentissage repose sur une propriété importante du modèle mixte (1.4) : il est associé à une fonction de vraisemblance strictement concave. Étant donné un ensemble de M échantillons $\mathcal{D} = \{X^{(m)} = (X_{\mathcal{C}}^{(m)}, X_{\mathcal{Q}}^{(m)}), m = 1, \dots, M\}$, la log-vraisemblance de $\Omega = (\Theta, \mu, \Delta, \Phi)$ est donnée par

$$\begin{aligned} \ell(\Omega : \mathcal{D}) &= \sum_{m=1}^{M} \log p_{\Omega}(X^{(m)}) \\ &= \sum_{m=1}^{M} \left[x_{\mathcal{C}}^T \Theta x_{\mathcal{C}} + \mu^T x_{\mathcal{Q}} - \frac{1}{2} x_{\mathcal{Q}}^T \Delta x_{\mathcal{Q}} + x_{\mathcal{C}}^T \Phi x_{\mathcal{Q}} \right] - \log Z_{\Omega}. \end{aligned} \tag{1.6}$$

On peut montrer que cette fonction est concave en Ω, et que le maxima est unique. Cette propriété permet d'identifier un unique modèle à partir des données.

Il est connu que les méthodes à base d'optimisation de vraisemblances ont tendance à overfitter les données d'apprentissage, et vont résulter en un réseau complètement connecté. L'utilisa-

tion de pénalisation permet de corriger ce problème. En l'occurrence, l'usage de la pénalisation $\ell - 1$ dite Lasso (voir Tibshirani [1996]) est légitime lorsque l'on recherche à apprendre un modèle parcimonieux. Cette pénalisation implique l'ajout d'un terme de régularisation dans la fonction à optimiser:

$$-\frac{1}{\beta}\sum_{i=1}^{k}|\omega_i| = -\frac{1}{\beta}||\Omega||_1,$$

où $||\cdot||_1$ est la norme L_1. Remarquons que ce terme est aussi concave.

Dans le cadre de notre étude, dans la mesure où certaines variables catégorielles ont été binarisées, la pénalisation ℓ_1/ℓ_2 est prescrite (voir Yuan and Lin [2006]) afin de tenir compte des apriori sur les variables issues de la binarisation. Cette pénalisation implique l'addition du terme suivant dans la fonction objective:

$$-\frac{1}{\beta}\sum_{j=1}^{J}||\omega_{K_j}||_2,$$

uù ω_{K_j} correspond au sous-ensemble $\{\omega_i, i \in K_j\}$, et où $\{K_j\}_j$ est l'ensemble des indices des variables issues de la binarisation. Observons que ce terme est aussi concave.

Toutes les pénalisations pénalisent les paramètres d'amplitudes élevées (positifs ou négatifs), cependant leurs impacts sont différents. De nombreux auteurs ont étudiés les impacts de leurs utilisations (e.g., voir Ng [2004]). En pratique, la principale différence réside dans le fait que les modèles appris avec une régularisation ℓ_1 ont tendance à être beaucoup plus parcimonieux que des modèles appris avec des régularisation ℓ_2 ou ℓ_1/ℓ_2, c'est-à-dire que ces modèles auront beaucoup plus de paramètres à zéro. D'un point de vue structurel, cela résulte en un graphe avec beaucoup moins d'arc et des potentiels plus parcimonieux.

Le problème d'apprentissage à résoudre est trouver l'estimateur

$$\hat{\Omega} = \underset{\Omega}{\text{Argmin}}\left(-\ell(\Omega : \mathcal{D}) + g(\Omega)\right), \tag{1.7}$$

où $\mathcal{D} = \{X^{(1)}, \ldots, X^{(M)}\}$ est l'ensemble d'apprentissage et g est une pénalisation sur les paramètres Ω du modèle. Dans la mesure où le Lasso tend à créer des structures parcimonieuses, on utilisera une régularisation particulière pour Δ afin d'assurer que cette matrice reste bien symétrique définie positive durant l'apprentissage; cette hypothèse est en effet obligatoire pour garantir que p_Ω est une densité valide, mais aussi pour des raisons numériques : assurer que la matrice Δ reste dans un compacte assure que la fonction d'apprentissage est gradient-Lipschitz, ce qui est une hypothèse requise pour les algorithmes d'apprentissage proposés par la suite. La régularisation envisagée est une contrainte d'ensemble compact définie comme suit : pour tout

$0 < \rho < 1$, on dénote par $\mathcal{K}_\rho$ le sous-ensemble compact des matrices définies positives défini par

$$\mathcal{K}_\rho = \{\, \Delta_0^{1/2}(I+\epsilon)\Delta_0^{1/2} \;:\; \epsilon \text{ is symmetric with} \\ -\rho < \lambda_{\min}(\epsilon) < \rho \,\},$$

où I est la matrice identité, $\lambda_{\min}$ et $\lambda_{\max}$ sont respectivement la plus petite et plus grande valeur propre et Δ_0 est la précision empirique définie par

$$\Delta_0 = \left[\frac{1}{M} \sum_{m=1}^{M} (X^{(m)} - \bar{X})(X^{(m)} - \bar{X})^T \right]^{-1},$$

où $\bar{X}$ est la moyenne empirique de l'ensemble $\{X^{(m)}, m = 1 \dots M\}$ de M échantillons. Ici, ρ est choisi arbitrairement pour assurer la convergence de l'optimisation numérique ; en pratique, on a simplement besoin de contrôler que l'estimateur obtenu reste bien à l'intérieur du compact.

La pénalisation utilisée est finalement

$$g(\Omega) = \lambda_\theta \sum_{k \neq k' \in K} \|\theta_{kk'}\|_2 + \mathbb{I}_{\{\mathcal{K}_\rho\}}(\Delta) + \lambda_\Delta \sum_{u<v \in \mathcal{Q}} |\Delta_{uv}| + \lambda_\Phi \sum_{k \in K, u \in \mathcal{Q}} \|\Phi_{ku}\|_2, \tag{1.8}$$

où $\mathbb{I}_{\{\mathcal{K}_\rho\}}$ est la fonction caractéristique de l'ensemble $\mathcal{K}_\rho$, i.e.,

$$\mathbb{I}_{\{\mathcal{K}_\rho\}}(\Delta) = \begin{cases} 0 & \text{if } \Delta \in \mathcal{K}_\rho, \\ +\infty & \text{otherwise,} \end{cases}$$

où $\theta_{kk'} = (\theta_{ii'})_{i \in k, i' \in k'}$ et $\phi_{ku} = (\phi_{iu})_{i \in k}$ où, pour tout $i \in \mathcal{C}$, k_i est l'ensemble des indices des variables crées lors de la binarisation. Remarquons que nous ne pénalisons pas les diagonales de Θ et Δ.

Le problème général d'apprentissage revient à trouver l'estimateur

$$\hat{\Omega} = \underset{\Omega}{\text{Argmin}} \left(-\ell(\Omega : \mathcal{D}) + g(\Omega) \right), \tag{1.9}$$

où $\mathcal{D} = \{X^{(1)}, \dots, X^{(M)}\}$ sont les données d'apprentissage, ℓ est une fonction de contraste et g est la pénalisation sur les paramètres Ω définie par (1.8).

Remarquons que la fonction à optimiser est strictement convexe, et qu'elle est la somme d'une fonction de classe $\mathcal{C}_1$ avec un gradient Lipschitzien, et d'une fonction non différentiable. L'algorithme du gradient proximal est particulièrement adapté à ce type de problème d'optimisation convexe. Il repose sur un schéma itératif, où à chaque itération, une évaluation de l'opérateur proximal de la fonction objective est réalisée. Cette évaluation implique la résolution d'un sous-

problème d'optimisation convexe, mais qui dans le cadre de notre étude, possède une forme explicite.

L'opérateur proximal $\text{Prox}_g : \mathbb{R}^n \to \mathbb{R}^n$ d'une fonction convexe propre $g : \mathbb{R}^n \to \mathbb{R}$ (c'est-à-dire convexe et à valeur dans la droite réelle $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$) est défini par

$$\text{Prox}_g(v) = \underset{x}{\text{argmin}} \left(g(x) + \frac{1}{2}||x - v||_2^2 \right),$$

où $||\cdot||_2$ est la norme L_2 usuelle. On rencontre souvent le cas où la fonction possède un multiplicande β, et l'opérateur proximal de βg est défini par

$$\text{Prox}_{\beta g}(v) = \underset{x}{\text{argmin}} \left(g(x) + \frac{1}{2\beta}||x - v||_2^2 \right). \tag{1.10}$$

Avoir une forme explicite de l'opérateur proximal permet un calcul rapide et amène à une vitesse de convergence rapide de l'algorithme du gradient proximal. Il y a de nombreuses situations dans lesquelles l'opérateur est explicitement connu, voir Parikh and Boyd [2013]. Plus spécifiquement, dans notre cas, les formules explicites des pénalisations rencontrées sont listées ci-dessous.

Dans le cas où g est la fonction caractéristique $\mathbb{I}_{\{\mathcal{C}\}}(\Omega)$ d'un ensemble convexe fermé $\mathcal{C}$, i.e.,

$$\mathbb{I}_{\{\mathcal{C}\}}(\Omega) = \begin{cases} 0 & \text{if } \Omega \in \mathcal{C}, \\ +\infty & \text{otherwise,} \end{cases}$$

la forme explicite de l'opérateur proximal est la projection orthogonale sur $\mathcal{C}$, i.e., la fonction $\Pi_{\mathcal{C}}(v)$ définie par

$$\Pi_{\mathcal{C}}(v) = \underset{\Omega \in \mathcal{C}}{\text{argmin}} ||\Omega - v||^2. \tag{1.11}$$

Dans le cas où g est une régularisation ℓ_1 $\beta\|\Omega\|_1 = \beta \sum_i |\omega_i|$, la forme explicite de l'opérateur proximal est l'opérateur de seuillage doux composante par composante s_β défini, pour chaque composant ω_i de Ω, par

$$s_\beta(\omega_i) = \begin{cases} \omega_i - \beta & \text{if } \omega_i \geq \beta, \\ \omega_i + \beta & \text{if } \omega_i \leq \beta, \\ 0 & \text{otherwise.} \end{cases} \tag{1.12}$$

Dans le cas où g est la régularisation ℓ_1/ℓ_2 $\beta \sum_j \|\omega_{K_j}\|_2$, la forme explicite de l'opérateur proximal est l'opérateur de seuillage doux composante par composante $\tilde{s}_{\beta,K}$ défini, pour chaque sous-ensemble de paramètres $\omega_{K_j} = \{\omega_i, i \in K_j\}$, par

$$\tilde{s}_{\beta,K}(\omega_{K_j}) = \begin{cases} \omega_{K_j} - \beta \frac{\omega_{K_j}}{\left|\left|\omega_{K_j}\right|\right|_2} & \text{if } \left|\left|\omega_{K_j}\right|\right|_2 > \beta, \\ 0 & \text{otherwise.} \end{cases} \tag{1.13}$$

Observons que dans (1.8), la matrice de précision Δ est pénalisée par la somme

$$\mathbb{1}_{\{\mathcal{K}_\rho\}}(\Delta) + \lambda_\Delta \sum_{u<v \in \mathcal{Q}} |\Delta_{uv}|,$$

pour laquelle il n'existe pas de forme explicite. On pourra alors utiliser l'algorithme *generalised forward-backward splitting* (voir Raguet et al. [2013]) pour résoudre le problème d'optimisation. En pratique, nous avons choisi d'utiliser une régularisation sans contrainte de compact et de contrôler le pas de gradient pour assurer la convergence.

L'algorithme du gradient proximal est explicité

Algorithme du gradient proximal déterministe pour minimiser $-\ell(\Omega : \mathcal{D}) + g(\Omega)$

Input Step sizes γ_t, a starting point $\Omega^{(0)}$,

At each i :teration t, given the current solution $\Omega^{(t)} = (\omega_1^{(t)}, \ldots, \omega_k^{(t)})$,
    Compute the gradient step $\tilde{\omega}^{(t)} = \omega^{(t)} + \gamma_t \nabla \ell(\omega^{(t)})$,
    Compute $\omega^{(t)} = \text{Prox}_{\gamma_t g}(\tilde{\omega}^{(t)})$.
Return the last estimation $\Omega^{(t)}$

Il existe de nombreux résultats théorique prouvant la convergence et assurant la vitesse de convergence (voir Parikh and Boyd [2013]). Notamment, l'algorithme converge avec des pas de gradient γ plus petits que $2/L$ où L est la constante de Lipschitz de ℓ. En pratique, cette constante n'est pas connue et on peu choisir les pas γ_t par un algorithme de linesearch, comme proposé par Beck and Teboulle [2009].

1.2.3 Apprentissage par gradient stochastique proximal

Dans cette section, on présente la méthode d'apprentissage d'un modèle en utilisant une version stochastique du gradient proximal. Rappelons que la fonction de partition (1.5) d'un modèle mixte (1.4) n'est pas calculable, et ne peut qu'être estimée.

On peut réécrire la densité mixte 1.4:

$$p_\Omega(X) = \frac{1}{Z_\Omega} \exp(\langle F, \Omega \rangle),$$

où Z_Ω est la fonction de partition (1.5), où $F = (F_1, F_2, F_3, F_4)$ est une statistique exhaustive de X avec:

- F_1 est la matrice indexée sur $\mathcal{C} \times \mathcal{C}$ définie par $F_1 = X_\mathcal{C} X_\mathcal{C}^T$,
- F_2 est le vecteur indexé sur $\mathcal{Q}$ défini par $F_2 = X_\mathcal{Q}$,
- F_3 est la matrice indexée sur $\mathcal{Q} \times \mathcal{Q}$ définie par $F_3 = -\frac{1}{2} X_\mathcal{Q} X_\mathcal{Q}^T$,
- F_4 est la matrice indexée sur $\mathcal{C} \times \mathcal{Q}$ définie par $F_4 = X_\mathcal{C} X_\mathcal{Q}^T$,

et où $\langle \cdot, \cdot \rangle$ est le produit scalaire défini par

$$\langle F, \Omega \rangle = \mathrm{Tr}\left(\Theta F_1^T\right) + \mu^T F_2 + \mathrm{Tr}\left(\Delta F_3^T\right) + \mathrm{Tr}\left(\Phi F_4^T\right). \tag{1.14}$$

On dénote par $\mathcal{D} = \{X^{(j)}\}_{j=1\ldots M}$ les données d'apprentissage constituées de M échantillons et $F^{(j)}$ leur statistique exhaustives. Avec ces notations, la log-vraisemblance des paramètres Ω étant donné $\mathcal{D}$ devient

$$\ell(\Omega : \mathcal{D}) = \frac{1}{M} \sum_{j=1}^{M} \langle \Omega, F^{(j)} \rangle - \log Z_\Omega. \tag{1.15}$$

On peut montrer que

$$\nabla \log Z_\Omega = \mathbb{E}_\Omega[F] = \mathbb{E}_\Omega[\mathbb{E}_\Omega[F|X_\mathcal{C}]]. \tag{1.16}$$

Cette formule amène à une estimation de $\nabla \log Z_\Omega$ par une méthode MCMC: si $\{\xi^{(m)}\}_{m=1\ldots\eta}$ sont η échantillons de p_Ω, alors

$$\nabla \log Z_\Omega \approx \frac{1}{\eta} \sum_{m=1}^{\eta} \mathbb{E}_\Omega[F|\xi_\mathcal{C}^{(m)}]. \tag{1.17}$$

Ces formules amène à une version stochastique de l'algorithme du gradient proximal.

1.2.4 Optimisation de la pseudo-vraisemblance

Dans cette section, on s'intéresse à l'optimisation de la pseudo-vraisemblance, une alternative à la vraisemblance classique. La pseudo-vraisemblance est introduite par Besag [1975], et repose

Estimation de $\nabla \log Z_\Omega$

Input Une paramétrisation Ω et une longueur de chaîne MCMC m,

1 Simuler η échantillons $\{\xi^{(m)}\}_{m=1\dots\eta}$ de la distribution $p_\Omega(x_\mathcal{C})$, en utilisant par exemple l'algorithme de Wolff 1,
2 calculer l'espérance conditionnelle $\mathbb{E}_\Omega[F|\xi^{(m)}] = (\mathbb{E}_\Omega[F_i|\xi^{(m)}],\ i = 1\dots 4)$ pour chaque échantillon $\xi^{(m)},\ j = 1\dots\eta$,
3 **Retourner** l'estimation $\nabla \log Z_\Omega$ donnée par

$$\nabla \log Z_\Omega \approx \frac{1}{\eta}\sum_{m=1}^{\eta} \mathbb{E}_\Omega[F|\xi_\mathcal{C}^{(m)}].$$

sur un estimateur consistent et facilement calculable.

Étant donné un jeu de données $\mathcal{D} = \{X^{(j)}\}_{j=1\dots M}$ de M échantillons et un modèle paramétré par Ω, la pseudo-vraisemblance est définie par

$$p\ell(\Omega : \mathcal{D}) = \frac{1}{M}\sum_{j=1}^{M} \log p_\Omega(X_\mathcal{Q}^{(j)}|X_\mathcal{C}^{(j)}) + \frac{1}{M}\sum_{j=1}^{M}\sum_{i\in\mathcal{C}} \log p_\Omega(X_i^{(j)}|X_{-i}^{(j)}), \tag{1.18}$$

où X_{-i} désigne toutes les variables de X sauf X_i.

Remarquons que notre définition de la pseudo-vraisemblance est légèrement différente de la pseudo-vraisemblance classique, qui est donnée par

$$\frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{n} \log p_\Omega(X_i^{(j)}|X_{-i}^{(j)}), \tag{1.19}$$

dans la mesure où nous distinguons les variables catégorielles et quantitatives, et que nous traitons la partie quantitative comme une vraisemblance conditionnellement gaussienne étant donné les variables catégorielles. En ce sens, notre pseudo-vraisemblance est plus proche de la vraisemblance classique que celle proposée par Besag [1975] et étudiée dans le contexte de l'apprentissage de modèles graphiques par Lee and Hastie [2015]. Observons aussi que notre pseudo-vraisemblance est définie sur un espace de paramétrisations tel que Θ et Δ sont respectivement symétrique et symétrique définie positive.

Observons aussi que la pseudo-vraisemblance (1.18) est une fonction strictement concave en Ω. En effet, elle est la somme d'un terme associé aux variables quantitatives et d'un terme associés aux variables catégorielles. On peut montrer que chaque terme est concave, et obtenir les formules suivantes des gradients.

Concernant la partie quantitative, étant donné $X_\mathcal{C}$, $X_\mathcal{Q}$ a une densité conditionnellement

gaussienne de moyenne $\Delta^{-1}(\mu + \Phi^T X_{\mathcal{C}})$ et de matrice de covariance Δ^{-1}. On a donc

$$\begin{aligned}\log p_\Omega(X_{\mathcal{Q}} \mid X_{\mathcal{C}}) = &-\frac{1}{2} X_{\mathcal{Q}}^T \Delta X_{\mathcal{Q}} + (\mu + \Phi^T X_{\mathcal{C}})^T X_{\mathcal{Q}} \\ &- \frac{1}{2}(\mu + \Phi^T X_{\mathcal{C}})^T \Delta^{-1} (\mu + \Phi^T X_{\mathcal{C}}) \\ &+ \log[(2\pi)^{-\frac{|\mathcal{Q}|}{2}} |\Delta|^{\frac{1}{2}}].\end{aligned}$$

En différentiant par rapport à Δ, Φ et μ, on obtient les gradients

$$\begin{aligned}\nabla_\Delta \log p_\Omega(X_{\mathcal{Q}} \mid X_{\mathcal{C}}) &= \frac{1}{2}\left[-X_{\mathcal{Q}} X_{\mathcal{Q}}^T + \Delta^{-1} + \Delta^{-1}(\mu + \Phi^T X_{\mathcal{C}})(\mu + \Phi^T X_{\mathcal{C}})^T \Delta^{-1}\right], \\ \nabla_\Phi \log p_\Omega(X_{\mathcal{Q}} \mid X_{\mathcal{C}}) &= X_{\mathcal{C}} X_{\mathcal{Q}}^T - X_{\mathcal{C}}(\mu + \Phi^T X_{\mathcal{C}})^T \Delta^{-1}, \\ \nabla_\mu \log p_\Omega(X_{\mathcal{Q}} \mid X_{\mathcal{C}}) &= X_{\mathcal{Q}} - \Delta^{-1}\mu - \Delta^{-1}\Phi^T X_{\mathcal{C}}.\end{aligned} \tag{1.20}$$

Concernant la partie catégorielle, la distribution conditionnelle des x_i sachant toutes les autres est donnée par

$$p_\Omega(x_i \mid x_{-i}) \propto \exp\left[\theta_{ii} x_i^2 + x_i \left(\sum_{j>i} \theta_{ij} x_j + \sum_{u \in \mathcal{Q}} \phi_{iu} x_u\right)\right].$$

Chaque variable x_i a donc une densité conditionnelle de Bernoulli, de moyenne

$$p_i = \mathbb{E}_\Omega[x_i \mid x_{-i}] = \frac{\mathrm{e}^{q_\Omega(x,i)}}{1 + \mathrm{e}^{q_\Omega(x,i)}},$$

avec

$$q_\Omega(x, i_0) = \theta_{i_0 i_0} + 2\Theta_{i_0,-i_0} x_{\mathcal{C},-i_0} + \Phi_{i_0,\mathcal{Q}} x_{\mathcal{Q}}, \tag{1.21}$$

où $x_{\mathcal{C},-i_0}$ désigne le vecteur $X_{\mathcal{C}}$ où l'entée d'indice i_0 a été enlevée, Θ_{-i_0,i_0} représente $(\theta_{i_0,j})_{j \neq i_0}$, la ligne d'indice i_0 de Θ sans l'entrée de coordonnée i_0, et $\Phi_{i_0,\mathcal{Q}}$ représente la ligne i_0 de Φ. On a donc

$$\begin{aligned}\log p_\Omega(X_{i_0} \mid X_{-i_0}) &= X_{i_0} \log P_\Omega(X_{i_0} = 1 \mid X_{-i_0}) \\ &\quad + (1 - X_{i_0}) \log(1 - P_\Omega(X_{i_0} = 1 \mid X_{-i_0})) \\ &= X_{i_0} q_\Omega(X, i_0) - \log(1 + \exp q_\Omega(X, i_0)).\end{aligned}$$

On obtient les gradients, pour tout $i, j \in \mathcal{C}$,

$$\nabla_{\Theta_{i,j}} q_\Omega(X, i_0) = \begin{cases} \mathbb{1}_{\{i=i_0\}} & \text{if } i = j, \\ \mathbb{1}_{\{i=i_0\}} X_j + \mathbb{1}_{\{j=i_0\}} X_i & \text{if } i \neq j, \end{cases}$$

et, pour tout $i \in \mathcal{C}$ et $v \in \mathcal{Q}$,

$$\nabla_{\Phi_{i,v}} q_\Omega(X, i_0) = \mathbb{1}_{\{i=i_0\}} X_v.$$

On en déduit que

$$\nabla_\Theta \sum_{i_0 \in \mathcal{C}} \log p_\Omega(X_{i_0} \mid X_{-i_0}) = \text{Diag}\left(E_\Omega(X, \mathcal{C}) \circ (2X_\mathcal{C} - 1)\right) - \text{Diag}\left(-X_\mathcal{C}\right) + 2X_\mathcal{C} X_\mathcal{C}^T - (E_\Omega(X, \mathcal{C}) X_\mathcal{C}^T + X_\mathcal{C} E_\Omega(X, \mathcal{C})^T), \quad (1.22)$$

où ici Diag (A) est la matrice diagonale de diagonale A, $A \circ B$ est le produit de Hadamard de A et B, et $E_\Omega(X, \mathcal{C})$ est le vecteur défini par

$$E_\Omega(X, i) = p_\Omega(X_{i_0} = 1 \mid X_{-i_0}) \frac{\mathrm{e}^{q_\Omega(X,i)}}{1 + \mathrm{e}^{q_\Omega(X,i)}}, \quad i \in \mathcal{C}.$$

De manière similaire, on obtient que

$$\nabla_\Phi \sum_{i_0 \in \mathcal{C}} \log P_\Omega(X_{i_0} \mid X_{-i_0}) = X_\mathcal{C} X_\mathcal{Q}^T - E_\Omega(X, \mathcal{C}) X_\mathcal{Q}^T. \quad (1.23)$$

Remarquons aussi qu'une autre manière d'écrire $q_\Omega(X, \mathcal{C}) = (q_\Omega(X, i))_{i \in \mathcal{C}}$ est de définir

$$q_\Omega(X, \mathcal{C}) = (\Theta + \Theta^T) X_\mathcal{C} + \text{Diag}\left(\Theta\right) \circ (1 - 2X_\mathcal{C}) + \Phi X_\mathcal{Q}.$$

Ces équations amène un algorithme d'apprentissage de structure utilisant une méthode de gradient proximal. Au contraire de la vraisemblance classique, tous les termes possèdent une forme explicite. On utilise donc une version déterministe du gradient proximal: si Ω_0 désigne le point initial de l'algorithme, et $\{\gamma_t\}$ une suite de pas de gradient positifs, alors étant donné $\Omega^{(t)}$, on déduit

$$\Omega^{(t+1)} = \text{Prox}_{\gamma_{t+1} g}(\Omega^{(t)} + \gamma_{t+1} \nabla p\ell(\Omega^{(t)}). \quad (1.24)$$

1.3 Détection et localisation d'anomalies

La détection d'anomalies (voir Chandola et al. [2009]) consiste à retrouver des éléments anormaux dans un jeu d'observations. La plupart des travaux en détection d'anomalies s'intéressent

à détecter des données (non conditionnellement) anormales, étant donné le reste du jeu de données. Dans cette section, on s'intéresse au contraire à de la détection d'anomalies conditionnelle (Chandola et al. [2009], Valko et al. [2011]), c'est-à-dire trouver des valeurs anormales dans un sous-ensemble de variables étant donné les valeurs des variables restantes.

Dans notre étude, on va définir les anomalies conditionnelles comme des changements de paramètres des lois conditionnelles des variables. On limite cette étude aux changements dans la moyenne conditionnelle des variables aléatoires du modèle : rappelons que les variables ont soit une densité conditionnellement gaussienne, paramétré par sa moyenne et variance conditionnelle, ou une densité de Bernoulli, uniquement paramétré par sa moyenne conditionnelle.

1.3.1 Détection d'un changement dans la moyenne conditionnelle

Supposons que l'on a déjà appris un modèle p_Ω de paramètres Ω et on note $x_1, \ldots, x_n$ les n variables du modèle. $\mathcal{D} = \{X^{(t)}, t = 1 \ldots M\}$ est un ensemble de données de test de M échantillons, indexés par le temps. Le problème de localisation est alors de trouver le sous-ensemble de variables $\{x_i, i \in 1 \ldots n\}$ pour lesquelles l'espérance conditionnelle $\mathbb{E}[X_i^{(t)}|X_{-i}^{(t)}]$, vue comme une fonction du temps, a changé par rapport à l'espérance conditionnelle $\mathbb{E}_\Omega[X_i^{(t)}|X_{-i}^{(t)}]$ du modèle appris $p_\Omega(X_i^{(t)}|X_{-i}^{(t)})$, où X_{-i} représente toutes les variables sauf X_i.

Basseville et al. [1993] propose une revue de nombreuses méthodes pour détecter des changements de paramètres dans des lois. Beaucoup de ces techniques reposent sur le calcul du rapport instantané de log-vraisemblance d'un échantillon X, défini par

$$s(X) = \log \frac{p_{\Omega_1}(X)}{p_{\Omega_0}(X)}.$$

Remarquons que, si $\mathbb{E}_{\Omega_0}$ et $\mathbb{E}_{\Omega_1}$ sont respectivement les espérances de X par rapport aux distributions p_{Ω_0} et p_{Ω_1}, et si s représente le rapport de log-vraisemblance, alors, si p_{Ω_0} et p_{Ω_1} sont des densités distinctes,

$$\mathbb{E}_{\Omega_0}(s) < 0 \quad \text{and} \quad \mathbb{E}_{\Omega_1}(s) > 0.$$

Cette propriété indique qu'un changement dans les paramètres Ω est caractérisé par un changement de signe de la moyenne des rapports de log-vraisemblance.

La littérature est très prolifique en détection de changement, et propose des approches pour de nombreuses situations, par exemple dans les cas où Ω_0 et Ω_1 sont connus, ou dans le cas où le temps t_0 auquel les paramètres changent est connu (voir Basseville et al. [1993] pour une revue exhaustive des méthodes de détection de changement). Parmi ces techniques, l'algorithme CUSUM Page [1954] a été introduit pour détecter séquentiellement des changement dans les moyennes des distributions, et repose aussi sur le calcul de ratios de log-vraisemblance.

Comme expliqué précédemment, sous l'hypothèse nulle $\Omega = \Omega_0$, la somme cumulée $\sum_{t=0}^{M-1} s_t$ a un drift négatif, et un drift positif sous l'hypothèse alternative $\Omega = \Omega_1$. L'algorithme CUSUM repose sur le calcul de la somme cumulée des parties positive du rapport de log-vraisemblance s. On défini la suite S_t, indexée par le temps, par

$$S_t = (S_{t-1} + s_t)^+ ,$$

où z^+ est la partie positive de z, c'est-à-dire $z^+ = \max(0, z)$ pour tout réel z. Remarquons que S_t reste proche de 0 sous l'hypothèse nulle, et adopte un drift positif après t_0. Le temps t_0 de changement de paramètres s'obtient en seuillant la fonction de décision S_t.

Dans nos travaux, nous avons adapté cet algorithme au problème de localisation d'anomalies dans un jeu de données $\mathcal{D} = (X_{\mathcal{C}}^{(t)}, X_{\mathcal{Q}}^{(t)})_{t=0,1,\dots}$. On suppose que le modèle de référence p_0 paramétré par Ω_0 a déjà été appris à partir de données normales, i.e., sans anomalies. Dans la mesure où on recherche un changement de moyenne conditionnelle qui peut être à la hausse ou à la baisse, on utilise le *two-sided CUSUM*, comme proposé par Basseville et al. [1993]. De manière similaire au CUSUM, pour tout t et pour chaque variable $X_i, i \in \mathcal{C} \cup \mathcal{Q}$, on défini le rapport instantané de log-vraisemblance conditionnel par

$$s_i^{(t)} = \log \left(\frac{p_{\Omega_1}\left(X_i^{(t)} | X_{-i}^{(t)}\right)}{p_{\Omega_0}\left(X_i^{(t)} | X_{-i}^{(t)}\right)} \right) ,$$

où $X_{-i} = \{X_j, j \in \mathcal{C} \cup \mathcal{Q}, j \neq i\}$ et p_{Ω_1} est la densité sous l'hypothèse alternative. On défini aussi récursivement une statistique de décision par $S_i^{(0)} = 0$ et

$$S_i^{(t)} = \left(S_i^{(t-1)} + s_i^{(t)} \right)^+ , \qquad t = 1, 2, \dots , \tag{1.25}$$

où $(z)^+ = \max(z, 0)$. Ici p_0 et p_1 correspondent respectivement aux densités sous l'hypothèse nulle, paramétré par Ω_0, et sous l'hypothèse alternative, paramétré par Ω_1, c'est-à-dire la densité conditionnelle du comportement anormal ciblé.

Jusqu'ici, on a supposé qu'on connaissait la densité alternative p_{Ω_1}. Cependant, Ω_1 ne peut pas être connu à l'avance, et nous devons maintenant trouver une manière de fixer$p_{\Omega_1} p_{\Omega_1}$. Dans la mesure où on s'intéresse à des rapports de vraisemblances conditionnelles, on va définir chaque densité conditionnelle $p_{\Omega_1}(x_i | x_{-i})$ séparément, en fonction du type de chaque variable x_i, i.e., catégoriel ou quantitatif.

1.3.2 La densité alternative pour les variables quantitatives

On s'intéresse aux variables quantitatives. On sait que la densité conditionnelle des variables quantitatives $X_{\mathcal{Q}}$ est la densité gaussienne multivariée $\mathcal{N}(\nu^{(t)}, \Delta^{-1})$, avec $\nu^{(t)} = \Delta^{-1}(\mu + \Phi^T X_{\mathcal{C}}^{(t)})$. On en déduit que, pour tout $i \in \mathcal{Q}$, la densité conditionnelle de $X_i^{(t)}$ étant donné $X_{-i}^{(t)}$ est la gaussienne univariée de moyenne

$$e_i^{(t)} = \mathbb{E}_\Omega[X_i^{(t)} \mid X_{-i}^{(t)}] = \Delta_{i,-i}^{-1}\Delta_{-i,-i}\left(X_{\mathcal{Q}-i}^{(t)} - \nu_{-i}^{(t)}\right) + \nu_i^{(t)}$$

et de variance

$$\sigma_i^2 = \mathrm{Var}_\Omega(X_i^{(t)} \mid X_{-i}^{(t)}) = \Delta_{ii}^{-1} - \Delta_{i,-i}^{-1}\Delta_{-i,-i}\Delta_{-i,i}^{-1}.$$

Remarquons que $e_i^{(t)}$ dépend des données, contrairement à σ_i qui ne dépend que des paramètres Δ.

On défini la densité alternative comme une gaussienne résultant de la translation de $\pm\delta\sigma$ de la densité conditionnelle sous l'hypothèse nulle. Le paramètre δ contrôle la sensibilité de la détection. Plus δ est large, plus anormal doit être le changement pour entrainer un drift positif de la statistique de décision S_t et donc pour être détecté. Plus δ est faible, plus le risque de fausse alarme est grand.

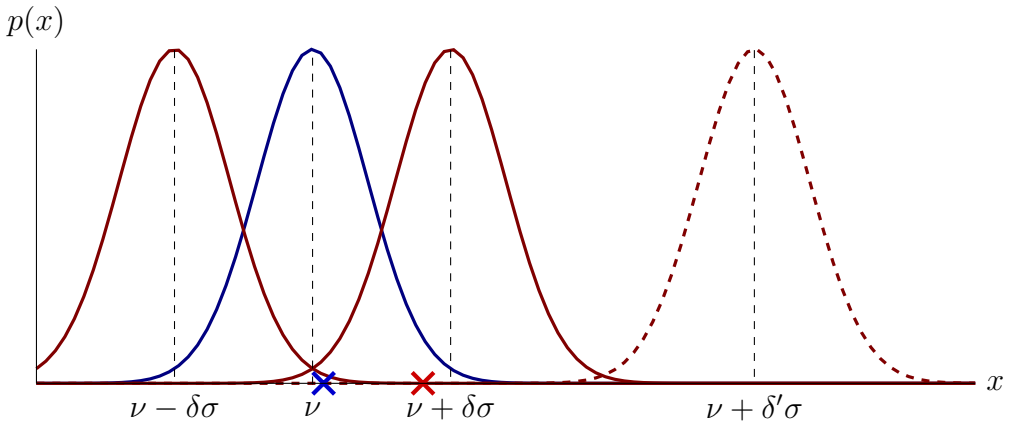


Figure 1.2: Densité alternative en dimension 1 pour une variable quantitative. Sous H_0, la densité conditionnelle normale est une gaussienne univariée (en bleu), et sous l'hypothèse alternative, la densité alternative (en rouge) est une translation de la gaussienne bleue de $+\delta\sigma$ ou $-\delta\sigma$, correspondant respectivement à un changement à la hausse ou à la baisse de la moyenne conditionnelle. La courbe en pointillé correspond au cas où l'alternative translatée de $\delta' > \delta$. Les croix correspondent à deux échantillons pour lesquels on veut décider s'ils sont simulés de la courbe bleue ou non. Intuitivement, quand l'alternative est la gaussienne rouge de moyenne $\nu + \delta\sigma$, on peut décider que la croix rouge est plus probablement issue de la courbe rouge que de la courbe bleue, quand la croix bleue est plus probablement issu de la courbe bleue. Au contraire, quand l'alternative est construite avec $+\delta'\sigma$, les deux croix sont plus probablement issues de la gaussienne bleue.

Ayant défini la densité alternative p_{Ω_1}, on peut expliciter le ratio de vraisemblance condition-

nelle. Pour tout $i \in \mathcal{Q}$, on a

$$\begin{aligned} s_i^{(t)\pm} &= \frac{1}{2}\left(\frac{X_i^{(t)} - e_i^{(t)}}{\sigma_i}\right)^2 - \left(\frac{X_i^{(t)} - (e_i^{(t)} \pm \delta\sigma_i)}{\sigma_i}\right)^2 \\ &= \pm\frac{(X_i^{(t)} - e_i^{(t)})}{\sigma_i}\delta - \frac{1}{2}\delta^2. \end{aligned} \tag{1.26}$$

En considérant $s_i^{(t)+}$ et $s_i^{(t)-}$, cette définition produit deux statistiques $S_i^{(t)+}$ et $S_i^{(t)-}$, pour détecter respectivement un changement à la hausse ou à la baisse de la moyenne conditionnelle $e_i^{(t)}$. Dans nos expériences, on considérera la somme $\bar{S}_i^{(t)} = S_i^{(t)+} + S_i^{(t)-}$ pour détecter un changement dans les deux directions.

Remarquons que le paramètre δ a une interprétation géométrique intéressante. En effet, le drift de la statistique de décision (1.26), sous l'hypothèse nulle, vaut $\mathbb{E}_\Omega[s_i^{(t)}|X_{-i}^{(t)}] = -\delta^2/2$. On voit que le drift est négatif et complètement contrôlé par δ.

1.3.3 La densité alternative pour les variables quantitatives

On s'intéresse maintenant aux variables catégorielles. Rappelons que chaque variable catégorielle $X_i, i \in \mathcal{C}$ a une densité de Bernoulli de moyenne

$$p_i = \mathbb{E}_\Omega[X_i \mid X_{-i}] = \frac{\mathrm{e}^{q_\Omega(X,i)}}{1 + \mathrm{e}^{q_\Omega(X,i)}},$$

where

$$q_\Omega(X, i_0) = \theta_{i_0 i_0} + 2\Theta_{i_0,-i_0}X_{-i_0} + \Phi_{i_0,\mathcal{Q}}X_{\mathcal{Q}}.$$

Spécifiquement pour les variables catégorielles, on défini la densité conditionnelle de l'hypothèse alternative comme une densité de Bernoulli de moyenne $a_i^{(t)}$. Le ratio de vraisemblance conditionnelles vaut ainsi

$$s_i^{(t)} = X_i^{(t)} \log\frac{a_i^{(t)}}{p_i^{(t)}} + (1 - X_i^{(t)}) \log\left(\frac{1 - a_i^{(t)}}{1 - p_i^{(t)}}\right). \tag{1.27}$$

Il y a deux manières de fixer le paramètre $\alpha_i^{(t)}$. Premièrement, on peut choisir $a_i^{(t)}$ de telle sorte que le drift la statistique de décision sous l'hypothèse nulle vaille aussi $-\frac{\delta^2}{2}$, comme c'est le cas pour les variables quantitatives. Ce drift est obtenu en calculant $\mathbb{E}_\Omega[s_i^{(t)} \mid X_{-i}^{(t)}]$. On obtient l'équation

$$p_i^{(t)} \log\frac{a_i^{(t)}}{p_i^{(t)}} + (1 - p_i^{(t)}) \log\left(\frac{1 - a_i^{(t)}}{1 - p_i^{(t)}}\right) = -\frac{\delta^2}{2}. \tag{1.28}$$

Il est facile de montrer que cette équation en $a_i^{(t)}$ (avec δ et $p_i^{(t)}$ fixés) possède deux solutions distinctes $a_i^{(t)+} \in [p_i^{(t)}, 1]$ (associée à la statistique $S_i^{(t)+}$) et $a_i^{(t)-} \in [0, p_i^{(t)}]$ (associée à $S_i^{(t)-}$), détectant respectivement un changement à la hausse ou à la baisse de la moyenne $p_i^{(t)}$, avec un drift négatif $-\frac{\delta^2}{2}$ sous l'hypothèse nulle (voir Figure 1.3). Pour les mêmes raisons que pour les variables quantitatives, on a considéré la somme $\bar{S}_i^{(t)} = S_i^{(t)+} + S_i^{(t)-}$ dans nos expériences.

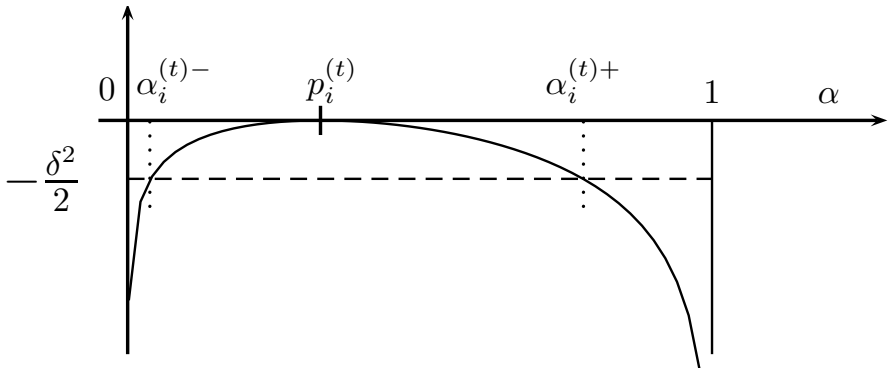


Figure 1.3: Evolution du drift négatif pour une variable catégorielle, vue comme une fonction de $\alpha \in]0, 1[$, où la moyenne $p_i^{(t)}$ de la densité de Bernoulli conditionnelle de x_i est fixée arbitrairement à $p_i^{(t)} = \frac{1}{3}$. Il y deux solutions, $\alpha_i^{(t)-} \in]0, p_i^{(t)}[$ et $\alpha_i^{(t)+} \in]p_i^{(t)}, 1[$ produisant un drift négatif de $-\frac{\delta^2}{2}$.

Une autre manière de choisir $\alpha_i^{(t)}$ est de fixer la variance conditionnelle $\mathrm{Var}_\Omega \left[s_i^{(t)} | s_{-i}^{(t)} \right]$ égale à un, comme c'est aussi le cas pour les variables quantitatives. On obtient alors aussi deux solutions $a_i^{(t)-}$ et $a_i^{(t)+}$, données par

$$\begin{aligned} a_i^{(t)-} &= \left[1 + \frac{1 - p_i^{(t)}}{p_i^{(t)}} \exp\left(+ \frac{1}{\sqrt{p_i^{(t)}(1 - p_i^{(t)})}} \right) \right]^{-1}, \\ a_i^{(t)+} &= \left[1 + \frac{1 - p_i^{(t)}}{p_i^{(t)}} \exp\left(- \frac{1}{\sqrt{p_i^{(t)}(1 - p_i^{(t)})}} \right) \right]^{-1}. \end{aligned} \tag{1.29}$$

La Figure 1.4 montre l'évolution du drift comme une fonction de $p_i^{(t)} \in]0, 1[$.

1.3.4 Notre version du two-sided CUSUM

Notre algorithme de localisation d'anomalies est présenté dans l'algorithme**??**.

Remarquons que cet algorithme est paramétré par deux scalaires:

- le paramètre de sensibilité δ,
- le seuil de détection h.

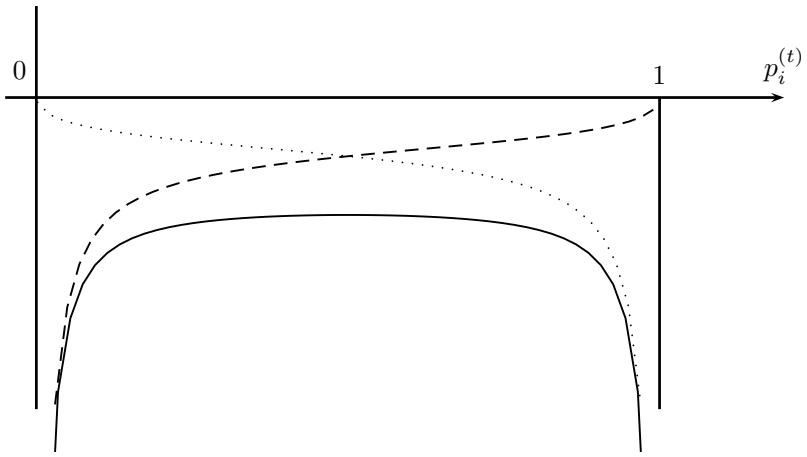


Figure 1.4: Évolution du drift négatif pour une variable catégorielle x_i quand les paramètres $a_i^{(t)+}$ et $a_i^{(t)-}$ sont données par (1.29), dans le cas où la variance conditionnelle $\mathrm{Var}_\Omega[S_i^{(t)}|s_{-i}^{(t)}]$ est égale à un. Le drift pour $S_i^{(t)-}$ est la ligne en points, le drift pour $S_i^{(t)+}$ est la ligne en tirets, et la ligne pleine est le drift pour $\bar{S}_i^{(t)}$.

Two-sided CUSUM pour la détection et localisation d'anomalies

Input Le modèle appris p_Ω, un paramètre de sensibilité δ, un seuil h, et un jeu de données $\{X^{(t)}\}_t$,

Initialiser $S_i^{(0)} = 0$ pour chaque $i \in \mathcal{C} \cup \mathcal{Q}$.
for each $X^{(t)}$, **do**
    for each $i \in \mathcal{Q}$, **do**
        Mettre à jour $S_i^{t+1} = (S_i^{(t)} + s_i^{(t)})^+$ en utilisant l'équation (1.26).
    for each $i \in \mathcal{C}$, **do**
        Calculer la moyenne $a_i^{(t)}$ de la Bernoulli alternative en utilisant l'équation (1.28) ou (1.29),
        Mettre à jour $S_i^{t+1} = (S_i^{(t)} + s_i^{(t)})^+$ en utilisant l'équation (1.27).
Pour chaque variable x_i, déterminer le temps de changement de paramètre $\tau_i = \min\{t; S_i^{(t)} > h\}$.

Retourner $\{\tau_i, i \in \mathcal{C} \cup \mathcal{Q}\}$.

1.4 Applications et résultats

1.4.1 Apprentissage d'un modèle à partir de données synthétiques

Dans cette section, on présente quelques expériences à partir de données synthétiques et de données réelles. Les données synthétiques sont générées à partir d'un modèle "en échelle" proposé par Lee and Hastie [2015]. La structure de ce modèle est présenté en Figure 1.5.

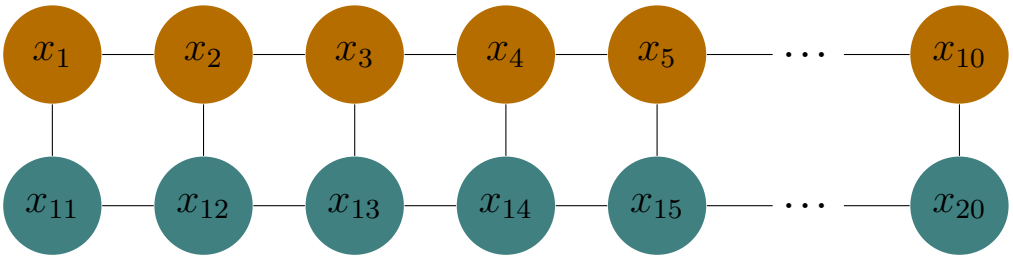


Figure 1.5: Structure du réseau en échelle utilisé pour simuler des données synthétiques. Ce réseau est constitué de 10 variables binaires $x_1, \ldots, x_{10}$ (en marron sur la couche supérieure) et 10 variables quantitatives $x_{11}, \ldots, x_{20}$ (en gris sur la couche inférieure).

Les paramètres $\Omega^* = (\Theta^*, \mu^*, \Delta^*, \Phi^*)$ de ce modèle ont été choisi de la façon suivante:

- Θ^* est une matrice carrée 10×10, avec -0.5 sur les entrées de sa diagonale et 0.5 sur les entrées des diagonales supérieures et inférieures,
- μ^* est un vecteur vertical composé de 10 coefficients nuls,
- Δ^* est une matrice carrée de dimension 10×10 avec 1 sur sa diagonale et 0.25 sur les diagonales supérieures et inférieures,
- Φ^* est une matrice 10×10 avec 0.5 sur sa diagonaleet 0 partout ailleurs.

Les paramètres de régularisation ont été choisis distincts pour chaque matrice Θ, Δ et Φ, et ont été choisi de telle façon à maximiser la vraisemblance sur des données de tests, parmi une grille de valeurs. En l'occurrence, dans le cas de l'optimisation de la pseudo-vraisemblance, les paramètres de régularisation ont été choisi tels que

$$\lambda_\Theta = 1.4\sqrt{\frac{\log(20)}{M}} \qquad \lambda_\Delta = 0.95\sqrt{\frac{\log(20)}{M}} \qquad \lambda_\Phi = 4.6\sqrt{\frac{\log(20)}{M}}, \tag{1.30}$$

et dans le cas de l'optimisation de la vraisemblance par gradient proximal stochastique,

$$\lambda_\Theta = 3.7\sqrt{\frac{\log(20)}{M}} \qquad \lambda_\Delta = 3.2\sqrt{\frac{\log(20)}{M}} \qquad \lambda_\Phi = 4.3\sqrt{\frac{\log(20)}{M}}. \tag{1.31}$$

On a en outre choisi des pas de gradients constants $\gamma = 1$ et $\gamma = 0.1$ respectivement pour la méthode du pseudo-vraisemblance et du gradient proximal stochastique (consécutivement des discussions de Atchade et al. [2015]). Des expérience d'apprentissage de structures sur ces deux méthodes ainsi que sur des techniques concurrentes actuelles de la littérature montrent que la méthode par pseudo-vraisemblance est meilleure que les autres approches, voir Figure 1.6.

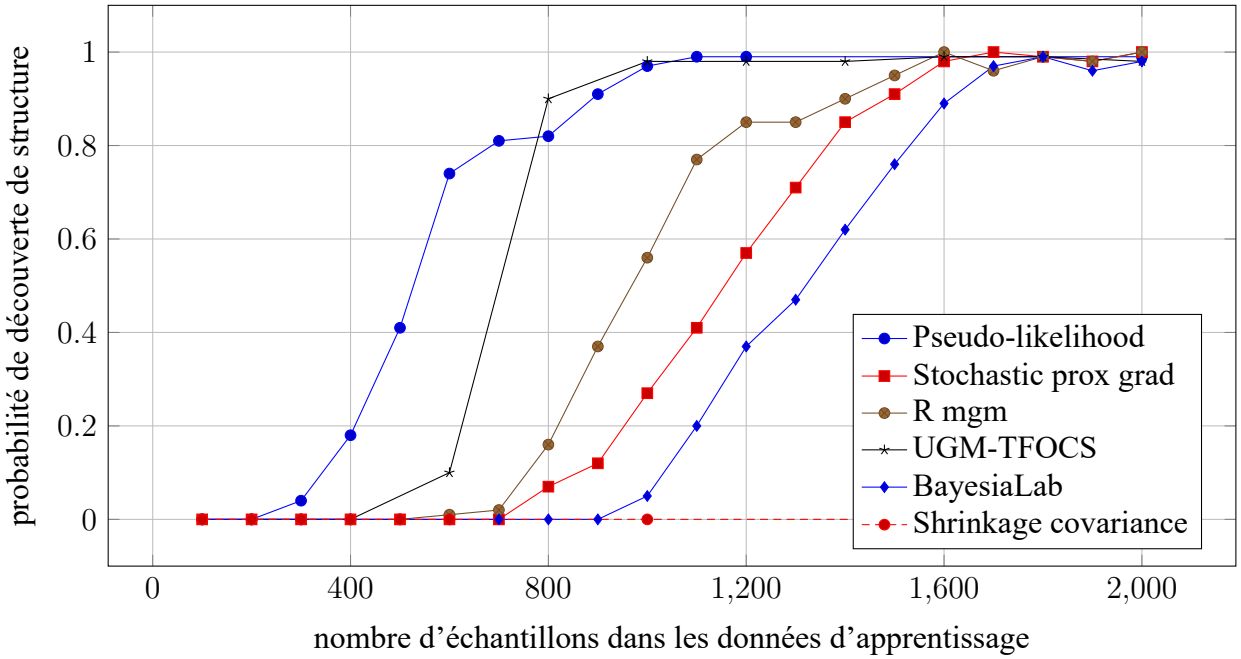


Figure 1.6: Probabilité de découverte de structure du réseau en échelle Ω^* ayant généré les données. On considère que la structure de Ω^* est découverte quand tous les arcs présents dans Ω^* ont été découverts et que aucun arc absent dans Ω^* n'a été découvert. Chaque point de ce graphe est la moyenne de 100 essais.

En outre, on peut aussi observer que la structure du graphe est apprise relativement vite, c'est-à-dire qu'elle n'évolue plus après quelques itérations de l'algorithme, en particulier pour l'optimisation de la pseudo-vraisemblance. La Figure 1.7 montre l'évolution des ratios de vrais positifs (TDR) et de faux positifs (FDR) durant l'apprentissage de deux graphes en échelles, le premier avec 20 variables et le second avec 200 variables (tous deux construits sur le même procédé qu'en Figure 1.5).

1.4.2 Détection et localisation d'anomalies dans des données synthétiques

On présente dans cette section les résultats en détection et localisation d'anomalies dans des données synthétiques. On suppose qu'on a déjà appris un modèle $\Omega*$ à partir de données normales. Les données de test sont composé de 50 échantillons normaux, simulés à partir de $\Omega*$, et de 50 échantillons simulés à partir de $\Omega*$ après avoir modifié un paramètre.

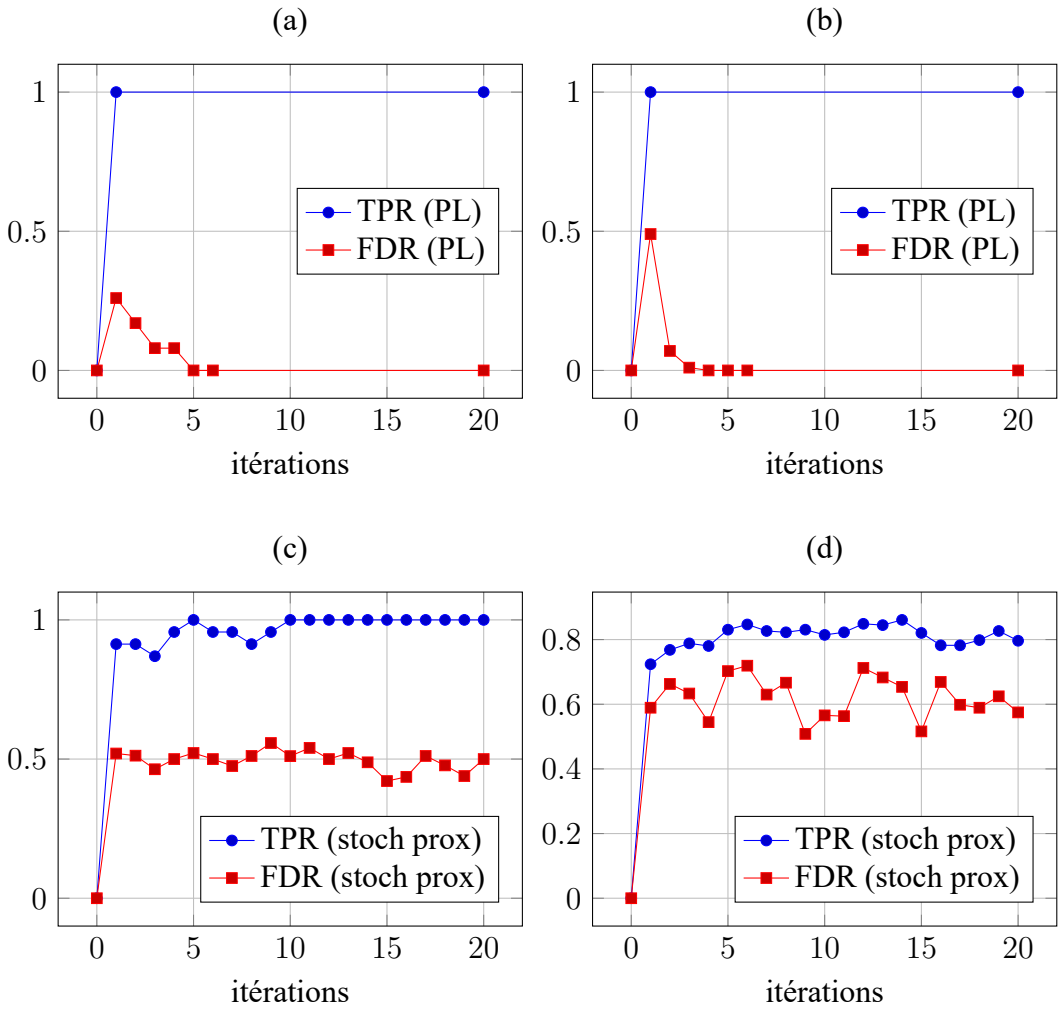


Figure 1.7: Évolution des taux de vrais positifs (TPR) et faux positifs (FDR) durant l'apprentissage d'un modèle. Les données sont simulées à partir d'un réseau en échelle semblable à celui de la Figure 1.5, avec 10 variables catégorielles et 10 variables quantitatives pour les graphes (a) et (c), et 100 variables catégorielles et 100 variables quantitatives pour les graphes (b) et (d). Pour chaque graphe, les données contiennent 1000 échantillons.

Le modèle $\Omega*$ est un modèle en échelle, présenté en Figure 1.8, dont les paramètres sont donnés par les matrices:

$$\Theta^* = \begin{bmatrix} -0.5 & 0.5 & 0 & 0 \\ 0.5 & -0.5 & 0.5 & 0 \\ 0 & 0.5 & -0.5 & 0.5 \\ 0 & 0 & 0.5 & -0.5 \end{bmatrix}, \qquad \mu^* = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\Delta^* = \begin{bmatrix} 1 & 0.25 & 0 & 0 \\ 0.25 & 1 & 0.25 & 0 \\ 0 & 0.25 & 1 & 0.25 \\ 0 & 0 & 0.25 & 1 \end{bmatrix}, \qquad \Phi^* = \begin{bmatrix} .5 & 0 & 0 & 0 \\ 0 & .5 & 0 & 0 \\ 0 & 0 & .5 & 0 \\ 0 & 0 & 0 & .5 \end{bmatrix}.$$

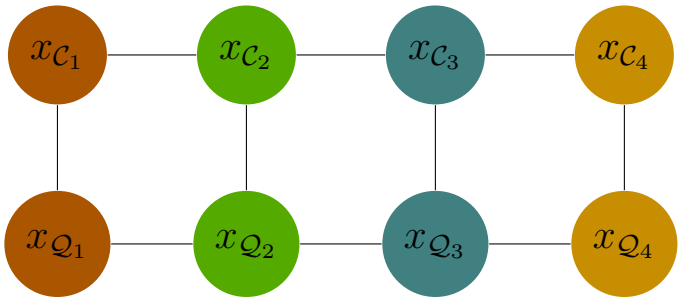


Figure 1.8: Structure du réseau utilisé pour les expériences. Ce réseau a quatre variables catégorielles $x_{\mathcal{C}_1}, \ldots, x_{\mathcal{C}_4}$ (sur la couche supérieure)et quatre variables quantitatives $x_{\mathcal{Q}_1}, \ldots, x_{\mathcal{Q}_4}$ (sur la couche inférieure).

On a testé trois différentes modifications sur des paramètres de Ω^*:

1. la distribution de la seconde variable quantitative (verte) est altérée en fixant μ_2^* à 3,
2. la distribution de la première variable catégorielle (marron) est altérée en fixant $\theta_{1,1}^*$ à -4,
3. la distribution conditionnelle de la première variable catégorielle (marron) et de la troisième variable quantitative (grise) est altérée en changeant $\phi_{1,3}^*$ à 2.

La Figure 1.9 présente l'évolution temporelle de la statistique $\bar{S}_i^{(t)}$ calculée pour chaque variable et pour les trois anomalies testées, en utilisant (1.29) pour les paramètres de la densité alternative des variables catégorielles, correspondant à une variance fixée à un. Comme attendu, les figures sur la ligne supérieure montrent que seule la statistique de décision correspondant à la variable quantitative verte possède un drift positif, indiquant que la variable quantitative verte porte seule un changement de paramètres. Les mêmes conclusions sont tirées sur les deux autres modifications sur $\theta_{1,1}$ et $\phi_{1,3}$. Ces résultats montrent que notre méthode détecte et localise correctement des changements de paramètres dans des distributions conditionnelles.

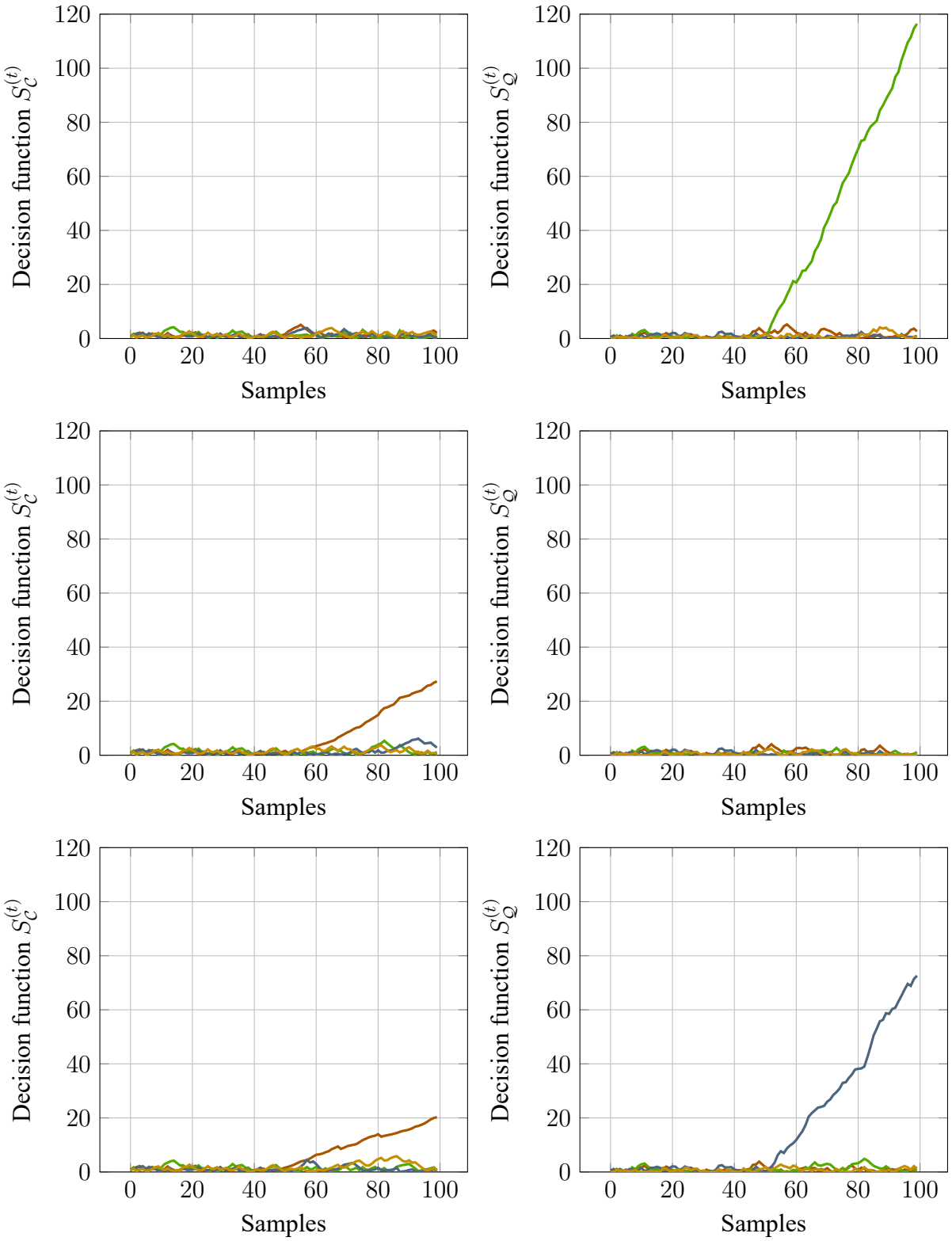


Figure 1.9: Evolution temporelle des statistiques $\bar{S}_i^{(t)}$ pour les variables quantitatives à gauche et pour les variables catégorielles à droites. Les couleurs des statistiques correspondent aux couleurs des variables de la Figure 1.8. Ligne supérieure : changement sur μ_2. Ligne centrale : changement sur $\theta_{1,1}$. Ligne inférieure : changement sur $\phi_{1,3}$. Pour chaque expérience, les 50 premiers échantillons sont simulés à partir de Ω^*, et les 50 derniers échantillons sont simulés à partir de la densité alternative.

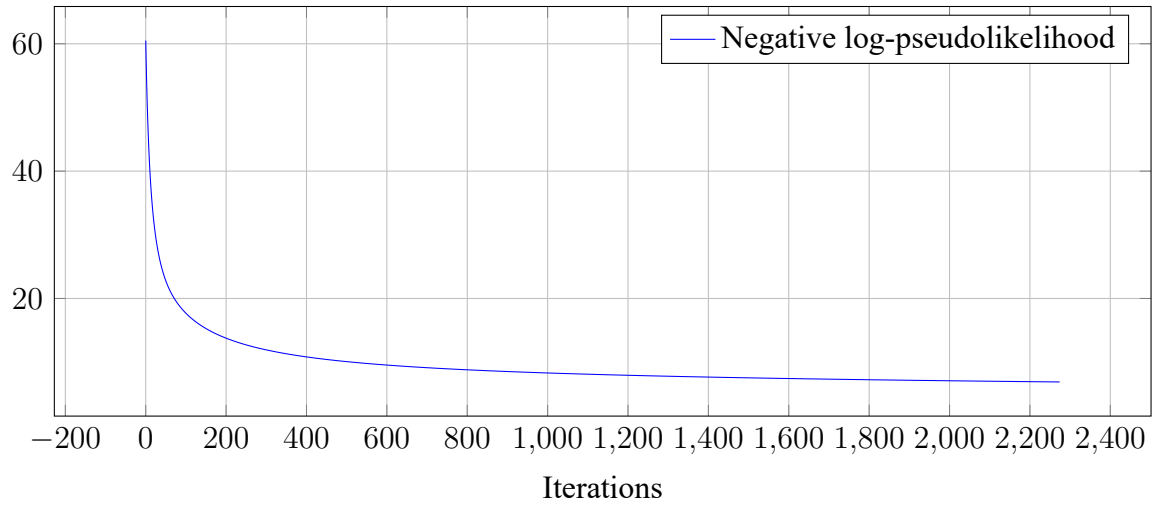


Figure 1.10: Optimisation de la psedo-vraisemblance durant l'apprentissage d'un modèle mixte, en utilisant la version classique du gradient proximal. Les données d'apprentissage sont issues du jeu de données *Touch and Go*.

1.4.3 Apprentissage d'un modèle et localisation sur données réelles

Les données réelles sont les données produites par le radar RBE2. On dispose de deux jeux de données, un jeu en faible dimension (appelé *Touch and Go*) composé $M = 1.5 \cdot 10^6$ échantillons de 77 variables binaires et 9 variables quantitatives, et un jeu en grande dimension (appelé *Tous modes*) composé de $M = 2.1 \cdot 10^6$ échantillons de 955 variables binaires et 49 variables quantitatives.

La Figure 1.10 et 1.11 montrent la décroissances des scores lors de l'apprentissage d'un modèle. La Figure 1.12 présente l'évolution de deux statistiques de décisions de deux variables du modèle *Touch and Go*, centrées sur une courte fenêtre temporelle autour d'une anomalie.

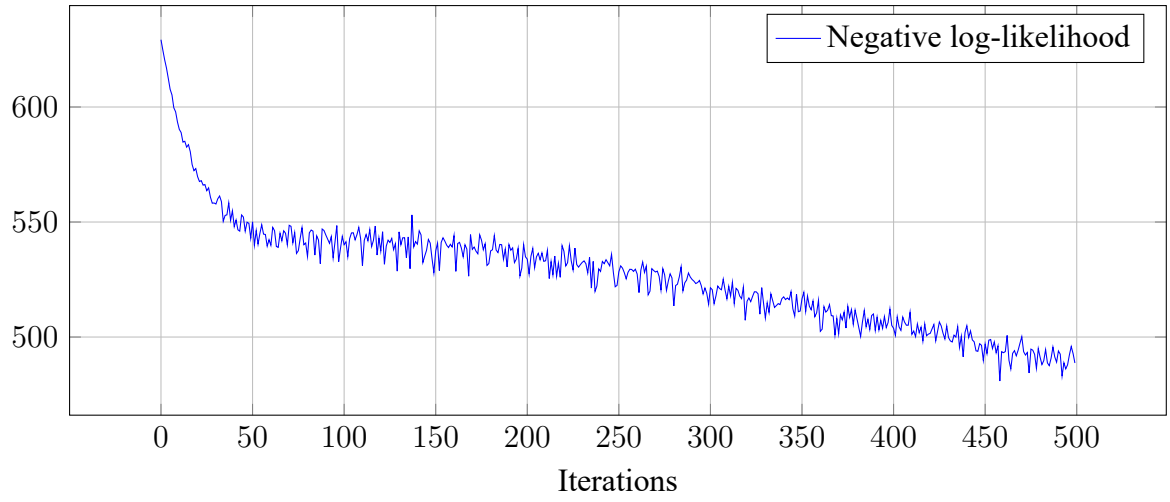


Figure 1.11: Optimisation de la vraisemblance durant l'apprentissage d'un modèle mixte, en utilisant le gradient proximal stochastique. Les données d'apprentissage sont issues des données *Tous modes*.

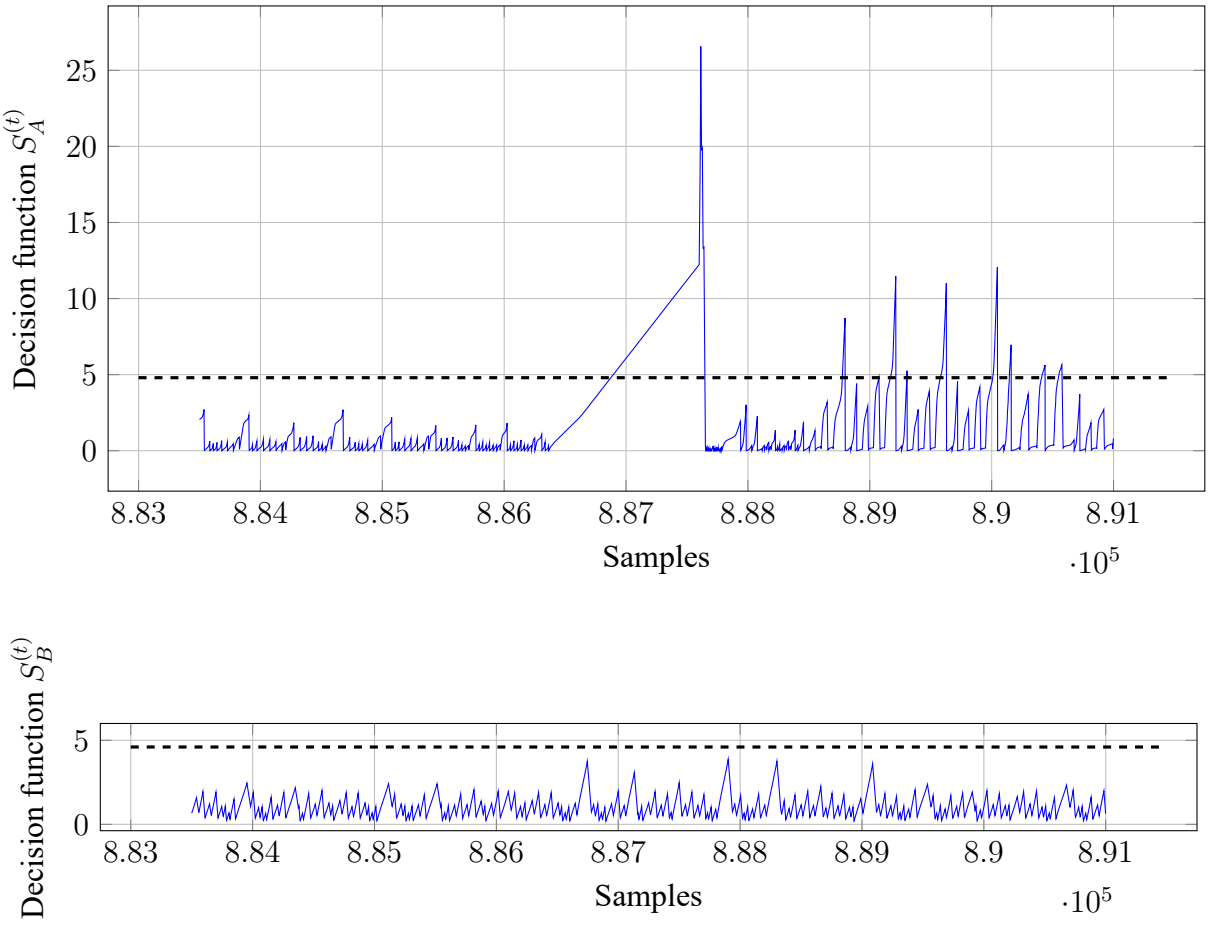


Figure 1.12: Statistiques de décision (1.25) $S_A^{(t)}$ et $S_B^{(t)}$ pour deux variables catégorielles A et quantitative B parmis les variables utilisées dans le scénario *Touch and Go*. Les données affichées correspondent à une fenêtre temporelle de 25 seconds (correspondant à environ 10 000 échantillons), centrée autour d'une anomalie portée par la variable A. La variable B n'est pas impliquée dans l'anomalie et sa statistique de décision ne présente pas de drift positif.

Chapter 2

Introduction

In this chapter, we introduce the anomaly detection problem that has motivated this thesis. Our work fits into a larger study made in Thales Airborn Systems that has started in 2010 when arose the first ideas of using a machine learning approach for completing the breakdown detection system that equips the radars produced in Thales. We will present the industrial problem, the concerned equipment, the data they produce during their use, what is the current detection system, and the industrial motivations that led to our study.

Thales is a worldwide group specialized in aeronautics, space, land transportation, security, and defence. Thales Airborn Systems, belonging to the aeronautic division, develops systems that answer many operational needs: embedded systems, sub-systems, complete systems or services, for military as well as civil customers. Within its range of expertise, this french company supervises many military programs and works as a subcontractor for other programs of other companies, like the Dassault Mirage and the Dassault Rafale, two french fighter jets produced by Dassault Aviation. More precisely for those two fighters, Thales develops and produces some of their main electronic features, among which combat radars, integrated defence-aids systems, flight management systems, fire control, interfaces for the pilots, pods, sensors, etc.

One of the most important systems of a fighter is its combat radar, often located in the nose of the plane. The Mirage 2000-5 fighter is equipped with a *RDY* (Radar Doppler Multitarget, see Figure 2.1), and the Rafale fighter is equipped with a *RBE2* (*Radar à Balayage Electronique 2 plans*, see Figure 2.2). Both radars can handle a lot of tasks, like air-to-air and air-to-ground scanning or targeting, though the RBE2 is a lot more advanced one. The RDY has a mechanical antenna that can only scan orthogonally to the surface of the antenna and has to rotate inside the nose of the aircraft in order to detect, track or engage targets. In contrast, the RBE2, which is a much more modern radar, has an electronic active antenna,

RDY

RBE2

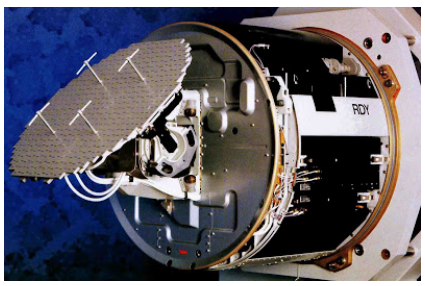


Figure 2.1: RDY

composed of hundreds of smaller antennas. The broadcast electromagnetic field is no more simply a cone perpendicular to the antenna, but rather a combination of all the fields created by the small antennas. By combining their effects, the RBE2 is able to detect targets in many directions at the same time, and keeping track of all of them, even during an engagement.

2.1 The built-in test

Both radars are an assembly of many components: in addition to the antenna, there is a hyper-frequency emitter, a modulator, some signal processing modules, a cooling system, an interface with the aircraft, and many other parts. All of those components are designed to work together in extreme situations, for instance when there are vibrations, brutal accelerations, high variations of pressures and temperatures, humidity or sand. These conditions can cause damages to the systems, or lead to an accelerated ageing. Thales thus has to ensure that the radars are working under such conditions, and also has to provide the tools that will make easy the maintenance and the repair when breakdowns occur.

Figure 2.2: RBE2

Built-in test

To address this issue, both the RDY and the RBE2 are both equipped with a *built-in test*. The built-in test has several objectives:

- Evaluate the working state of the radar and inform the other systems of the plane,
- Detect and localise the broken down elements during the mission to make the online repairing easy,
- Produce exploitable reports for offline repairing.

To achieve those goals, the built-in test is based on three major activities:

Raw data

- Collecting the *raw data* produced during integrated tests in the radar,
- Process those pieces of information in order to detect and localise the failures,
- Publish all the necessary information to inform the plane about the working state of the radar.

2.1.1 The current breakdown detection system

Internal communication bus

The tests used by the built-in test are either starting tests (only made when the radar is starting or after a reset), permanent tests (either made periodically or after a breakdown) or chain tests. The collected breakdown information from these tests are broadcasted on the Internal Communication Bus (ICB) of the radar toward the *maintenance manager* using specific frames (called CRD). On the RBE2, the conception of the ICB goes back to the early nineties. The RDY has a similar functioning, but has not been studied in this thesis.

Maintenance manager

Several kinds of breakdowns can occur, from fugitive breakdowns to material destructions. The appearance of such major breakdowns is watched by some security devices that will trigger immediate appropriate responses to protect the equipment. On the other side, minor anomalies are *filtered* by the maintenance manager and often ignored. If the breakdowns persist, the concerned components are rearmed and the radar is reset. If after that reset the radar works normally, the radar continues its operational functioning. In the other case where the component is still not working, it is declared having broken down and a CRD is sent to the plane.

Filtering

The tests that the built-in test uses to check the working state of the radar can be seen as simple if-then-else rules, written by the radar experts over the years. The maintenance manager constantly receives frames that contain tests reports and decides to filter or not the breakdowns that do not compromise the material security of the radar. Through that filtering process, the maintenance manager updates counters, by incrementing them when a breakdown is detected, or decrementing them when no breakdown is seen after a short period of time. As long as these counters remain under fixed thresholds, the radar continues its working mode, and no specific breakdown report is broadcast to the plane. However, if one of these thresholds is reached, the radar engages a reset process.

Example 2.1 False alarm raised by the maintenance manager The temperature of a lot of components is constantly measured and has to remain within some interval. For instance, the temperature of the small antenna's components are acceptably lower than 30 degrees when the antenna is switched off, and is supposed to be much higher only when it is on. If a too high temperature is measured when the antenna is on, an anomaly might be detected. However, the definition of a too high temperature depends on the context: if the antenna has been shut down and restarted within some seconds, it is normal to see a high temperature.

Example 2.2 Non-detection due to uncommon correlations A radar that just came out the production lines presented a strange anomaly during operational tests: the radar showed to the pilot many targets on its screen, when there was nothing in the sky. Although it was clearly an anomaly, no alarm was ever raised because this situation has never been anticipated as a possible breakdown, and the technicians had to manually investigate to solve that case. It took several

months to a whole expert team to understand the issue: due to assemblage mistake of a high-frequency waveguide, a component of the radar was disrupted, which led to the anomalous plots on the pilot screen.

The use of rules and filtering to analyse the raw data might lead the maintenance manager to raise false alarms or have non-detections. The example 2.1 illustrates a false alarm issue, and the example 2.2 illustrates a non-detection issue by showing how complex it can be to anticipate uncommon anomalies.

During operations, the radars are used under configurations that depend on the mission: climatic conditions, kind of missions, duration, etc. During the production, such conditions are reproduced artificially to test the radar. There are special rooms where the climatic conditions can be controlled and test benches to which the radar is connected for the test. One of the roles of these test benches is to reproduce the stimuli that the radar would receive during real operations, by executing *deterministic scenarios*. There is a finite number of scenarios, which are designed to guarantee the working state of the radar. The example 2.3 and the example 2.4 describe two real scenarios used for the tests, and that will serve as contexts for the experiments in section 5.

Deterministic test scenarios

Example 2.3 The scenario *Touch and go* The simplest scenarios is the *Touch and go*. This scenario runs through several landings and take-offs, what is a classic exercise for planes and pilots, even in non-military contexts. During this scenario, only a few parts of the radar are tested, like the antenna or the cooling system, whereas others are off, like the signal processing systems.

Example 2.4 The scenario *All modes* The scenario *All modes* is the richest and most complete one. The aim of that scenario is to test the transition between every working mode to all the others modes. Among those modes, we can find fire-control, combat, passive surveillance, target tracking, ground scanning, very low altitude, etc. This scenario may be executed under several environmental conditions: fog, humidity, heat, sand, wind, etc. Every part of the radar is consequently tested during the modes rotation, producing the biggest data file among all scenarios.

Either after a mission of after a test during production, the breakdown reports are analysed offline by the radar expert to investigate further the breakdowns. Although the raw data of the radar are extractable, they are never directly analysed to locate breakdown, but only used to clarify misunderstood breakdowns. This way of working is optimized for an operational use, but might miss some breakdowns (as illustrated in Example 2.2), and will anyway ignore minor breakdowns that will never be investigated, since they won't be reported in the CRDs.

2.1.2 The data circulating on the ICB

As explained above, all the components of the radar are communicating together using the ICB, an internal common communication bus. On that bus are circulating raw data that are analysed by the maintenance manager, as well as breakdown reports only sent when a major breakdown occurs. This reports contain a synthesis of the detected breakdown and its detected causes. The following will explain how these data is produced and sampled.

2.1.2.1 The structure of the data

The raw data sent by the components follow a very strict communication protocol. The support for the exchange of data on the ICB are *frames*, each of which carries between 12 bytes and 32 bytes. At a lower level, the frames are composed of several groups of two bytes called *words*, and each frame thus contains between 6 and 12 words. Frames are containers that always carry the same information – like gains, temperatures, states, etc – and are broadcasted on the ICB by the same sender to the same addressee.

Frame

Word

At a higher level, the frames exchanged on the ICB are grouped to form bigger structures called *messages*. The messages currently used contain between 1 and 138 frames. Considering a message instead of considering individually the frames it carries make the communication protocol lighter: instead of querying each frame separately, a single query is used for the whole message. Thus, either all the frames of a message are sent – one by one – or no frame of this message is sent. Since queries are also frames broadcast on the ICB, regrouping frames into messages considerably reduces the traffic load on the communication bus.

Message

It is important to note that this decomposition into frames and words is a very low-level data structure; words, frames and messages do seldom correspond to interpretable pieces of information, as physical measures like a gain, or categorical information like a working state or a mode. Such pieces of information will be later referred to as *fields*. A word may actually be the concatenation of several fields, and a field that requires 32 bits to be encoded will be stored on two words. Generally speaking, the number of fields in a frame varies from 4 to 500, with a mean around 100. The Figure 2.3 illustrate this data structure, and the Example 2.5 give an example of a true message used on the ICB.

Fields

Example 2.5 Content of the message AGVE The message AGVE is sent by the active antenna, headed to the maintenance manager. It is composed of three frames, each frames being composed of 16 words. This message carries several kinds of information, like ICB signatures, dates or working states. Its emission on the ICB is thus split into three sendings. Note that there is no guarantee that the emission of other frames will not be inserted between two frames of the AGVE message.

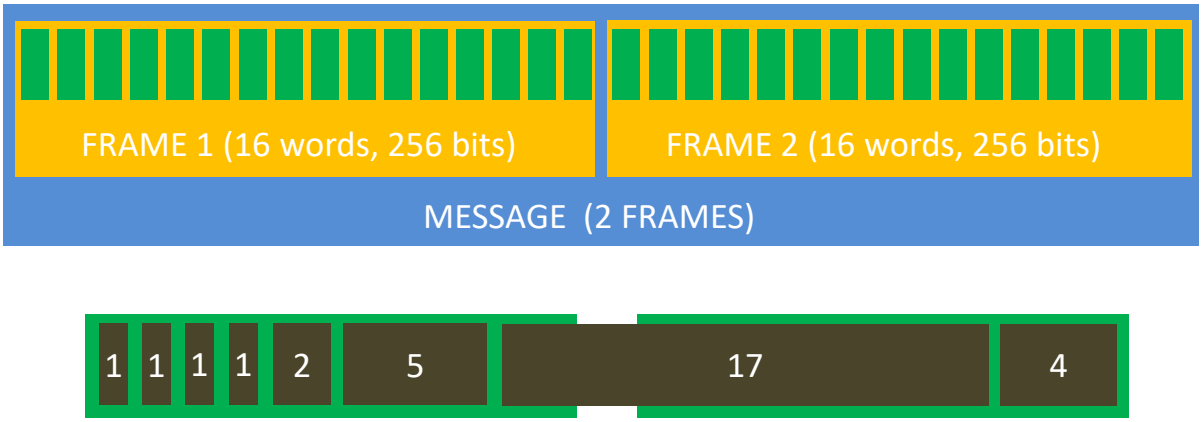


Figure 2.3: Scheme of the structure of an arbitrary message and the information it carries. At the top is shown the decomposition of a message (in blue) into two frames (in orange) and the decomposition of the frames into words (in green). On this figure, the message is composed of two frames, where each frame contains 16 words of 2 bytes. At the bottom is shown how fields (in brown) are stored in words: the words have themselves no real meaning, and the fields (the number of bits on which they are encoded is written in white) are either concatenated or split to fit into the 2 bytes of each word: one information might be encoded on 1 or 2 bits in one frame, or exceed to another frame if it is too big.

2.1.2.2 The different samplings of the messages

In addition to the different kinds of message structures, each frame has a specific sampling process. These samplings are related to the different tests that the built-in test runs during the use of the radar (see subsection 2.1.1). In this study, we have met three sorts of sampling:

Periodic sampling: Some messages are broadcast at a fixed frequency, from around one emission every few milliseconds for the most frequent, to one emission every second for the least frequent.

Contextual sampling: Some messages are broadcast only when a specific event is occurring, like it is the case when the built-in test has detected a breakdown. This kind of messages may never be sent if such an event never occurs.

Irregular sampling: Some messages have a very irregular sampling: the frames can be emitted at a high frequency for a short period of time, and then be quiet for seconds. The sampling of such frames is a very low-level process and is not the concern of this study, since we will remove all of those frames when processing the data.

This mixture of sampling is quite complex but has been designed by the experts to investigate breakdowns more rapidly: it is indeed possible to modify the content of the frames or even add

new ones, and that operation can be done quite often to adapt the data production to the expert needs.

2.1.2.3 The acquisition file

There is a card of the RBE2 that records some of the messages that circulate on the ICB, in order to allow an offline analysis of the built-in test breakdown reports and the raw data. The choice of which message to record depends on the needs of the experts and on the evolution of the message list. The RBE2 is currently using more than 8 000 different words, corresponding to around 100 000 fields.

Acquisition file

Whenever a frame is sent on the ICB, that frame is caught and written in an *acquisition file* (in a binary format). This acquisition file is thus composed of the succession of the records of all the frames broadcasted on the ICB, in the order of which they have been read by the recording card. Each record contains the content of a frame, plus the timestamp of when the frame has been received by the recording card. The figure 2.4 shows a text version of the first lines of an acquisition file.

```
1  b88b,321886261,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff
2  b886,321886406,f1ff,ffff,ffff,fffd,7fff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,0000,0000,0000,0000
3  b88b,321911223,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff
4  b88b,321936550,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff
5  b88b,321961245,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff
6  b88b,321986262,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff
7  b88b,322011271,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff
8  b88a,322036186,0000,0000,0000,2026,100f,1011,0000,001a,0000,00c9,0000,0000,0000,0000,0000,0000
9  b88b,322036370,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff
10 b88b,322061248,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff
11 b88b,322086254,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff,ffff
```

Figure 2.4: First lines of an acquisition file in an ASCII format. Each line corresponds to a frame recorded by the recording card. The first column indicates which frame has been stored on each line, the second column is the timestamp of the record, and the content of the frames is stored in the remaining columns in the green zone. In this green zone, the comas are separating the several words of the frame, written with hexadecimal hexadecimal numbers – corresponding to two bytes. As it can be seen here, the real fields carried by the frame (temperatures, gains, states, …) are not yet visible and require more processing to be accessible.

At this point, it is important to note that the data produced by the radar during the tests are random in the way they are produced, even if the scenarios are deterministic: it is not possible to simply compare the results files of a test with a reference file and look for the differences. Even one radar running twice the same scenario in the same environment will produce two different acquisition files. Fluctuations, external or internals disruptions, execution times, synchronisa-

tions are among the plentiful causes that will create differences between two runs, even with the same radar and the same scenario.

The resulting acquisition files will have a different number of lines, with a different order of received messages, with potentially different frame's content. However, when no breakdown occurs, the behaviour of the radar remains the same, and the breakdown detection system we will develop will have to deal with this file variability. As explained in the next sections, we have thus concentrated our study on a probabilistic approach.

2.2 Industrial problem of production stage

2.2.1 Motivation for a machine learning approach

As explained above, the built-in test is designed to be optimized for operation, by providing an online breakdown detection tool through the filtering processes of the maintenance manager, and producing breakdown reports for offline investigations by the experts.

However, the whole data recorded on the ICB is never fully analysed directly by the experts: currently, there isn't a tool that detects and records the breakdowns ahead of the maintenance manager. An expert that would like to investigate for unreported breakdowns – which are mainly fugitive breakdowns – would have to manually display the value taken by each of the 8 000 words and look for anomalous values. Though it would be fairly daunting, this way of looking for unrecorded breakdown would maybe work for anomalous values of quantitative fields (which may be far different than normal values), but is definitely not worth considering it for binary or categorical fields, which values depend largely on the values of the other fields and whose anomalous values are almost never visually identifiable.

If the notion of a breakdown is clearly defined in the built-in test by thresholds of counters of anomalies detected by the maintenance manager, the definition of a breakdown in raw data is not fixed.

It is clearly not possible to list all the potential breakdowns, because the number of anomalous situations is exponential in the number of words. On the other hand, in production, since the tests consist in the execution of deterministic scenarios, the global behaviour of good radars should always be the same, if the set of used words and their sampling isn't changed.

2.2.2 Reformulation of the anomaly detection problem

The industrial need consists in a breakdown detection tool that can complete the current detection system – formed by the raw data filtering of the maintenance manager and the radar experts investigations of the breakdown reports – by using a statistical approach to directly analyse the

raw data and detect breakdowns. That tool is thus not intended to supplant the maintenance manager or to interfere in the expert management of the ICB and the way the different components of the radar are communicating between each other: it will use the acquisition file produced during the test, and analyse it offline.

Since it is not possible to list all possible breakdowns, the notion of breakdown has to be redefined. From now on, we will speak about *anomalies* rather than breakdowns. An anomaly is the emission of a frame on the ICB, which field's values are not consistent with the normal behaviour of the radar. This definition of this normal behaviour depends on the scenario played during the test.

Anomaly

However, just detecting at what time an anomaly occurs isn't helpful for the Thales experts. An anomaly might be detected after the emission of a specific frame, but that frame might only be an anomaly trigger and not be the root cause of the anomaly: without breakdown synthesis report, the expert will have in any case to investigate the raw data to find the true cause of the anomaly.

In this thesis, we will explain the (industrial) *localisation problem*, what we define as finding which component(s) of the radar is the cause of the detected anomaly. From an industrial point of view, there are two main problems to solve :

Localisation problem

1. using the acquisition file produced during the execution of every test scenario, what we will refer later to as the *reference file*, learn the normal behaviour of a radar;

Reference file

2. compare newly recorded data from the execution of a known scenario, what we will refer later to as the *test file*, to the normal behaviour of this scenario, in order to detect anomalies and locate the components that caused these anomalies.

Test file

In addition to this two objectives, the solution will have to address some challenges specified by the final industrial usage:

1. the fields carried by the frames are heterogeneous: some are quantitative and represents physical quantities which possess an arithmetic, whereas other are categorical and with which no mathematical operations can be used,

2. the anomaly detection and localisation method will be used on a standard test bench which has the power of a classic working computer (CPU with around 3 Ghz and 8 Gb of RAM),

3. the execution of the anomaly detection and localisation algorithms will require at most the same time as the execution of a scenario, which is around two hours.

Before the presentation of our anomaly detection and localisation solution, we will explain how we are processing the acquisition file.

2.3 Adapted data preprocessing

Before being used by any software or algorithm, the data need to be extracted from the acquisition file, which is, as explained in the section 2.1.2.3, a binary file which contains the succession of all the frames emitted on the ICB, in the reading order of the recording card. There are several objectives of this acquisition file processing:

- understanding how the acquisition file format and how to translate it into the chosen data file format,
- defining a file format that is suitable for a lot of machine learning libraries or software.

This targeted data file format will simply be a file where each line corresponds to a state of the radar – in some way related the receiving of a frame – and each column corresponds to a field (single information like a physical measure, a working mode, status, …), what we will refer to as a *complete data file*. However, the number of fields in a frame is very low compared to the total number of fields, with a ratio of less than 0,1%. The resulting file will thus have a lot of missing values. To address that issue, we have first reduced the number of used field. The choices we made to that end are presented in section 2.3.1. Secondly, we have chosen a specific strategy to deal with the missing values. The motivations and consequences of that choice are explained in section 2.3.2.

Complete data file

2.3.1 Defining a dimension reduction strategy

In this section, we present the strategy we deployed to reduce the number of frames we are keeping from the acquisition file during its processing.

As said before, the number of frames used by the built-in test is around 400, which corresponds to several hundreds of thousands of fields. Without any field reduction strategy, the resulting file contains around 100 000 columns and several tens of millions of lines, corresponding of all the frames broadcasted on the ICB. This file would size around one Terabyte on a hard drive, what would make it impossible to use with the current computer configurations we have.

It appears we can get rid of a lot of frames, because they are redundant with other frames, contain useless information for our study like dates or counters, or are already synthesized in other frames. After collecting the expert thoughts on that question, we have reduced the number of used frames from around 400 to a few tens, depending on the scenario. This reduction of used frames is also decreasing the number of lines in our data file, since we won't consider the emission of not kept frames. This results in a 90% reduction of the number of lines.

For reasons that will become clear in chapter 3, we won't also keep fields that are constant in the reference file. After having removed the constant fields, the final number of kept fields

for the rest of this study varies from 69 (corresponding to the *Touch and go* scenario presented in example 2.3) to 125 (for the *All modes* scenario of the example 2.4), depending on which scenario is used during the tests.

2.3.2 Production of a data file

Variable

First, we introduce the notion of *variables* that will replace the notion of fields in the following. As it will be more explained in the chapter 3, our algorithms won't be able to deal with categorical non-binary fields. These fields will be removed in favour of binary variables following the classic 1-to-K encoding scheme (see section 3.1.1). The definition of a variable thus depends on the type of the field it corresponds to: for quantitative fields – like physical measures – and categorical binary fields – like bits of working, variables and fields are interchangeable. But each categorical non-binary field is transformed into as many binary variables as the number of values it takes. Note that the 1-to-K encoding scheme can drastically increase the number of variables, if a categorical field takes a lot of different values in the reference file. Again, that increase depends on the scenario: for the *Touch and go* scenario there is no non-binary categorical variables, so the number of variables and the number of fields are the same, but for the *All mode* scenario, the number of variables is 839, where the number of fields was 150.

Once the field reduction strategy is defined, we can reassemble the text version of the acquisition file to a data file where each line is an instance of all variables, without missing values.

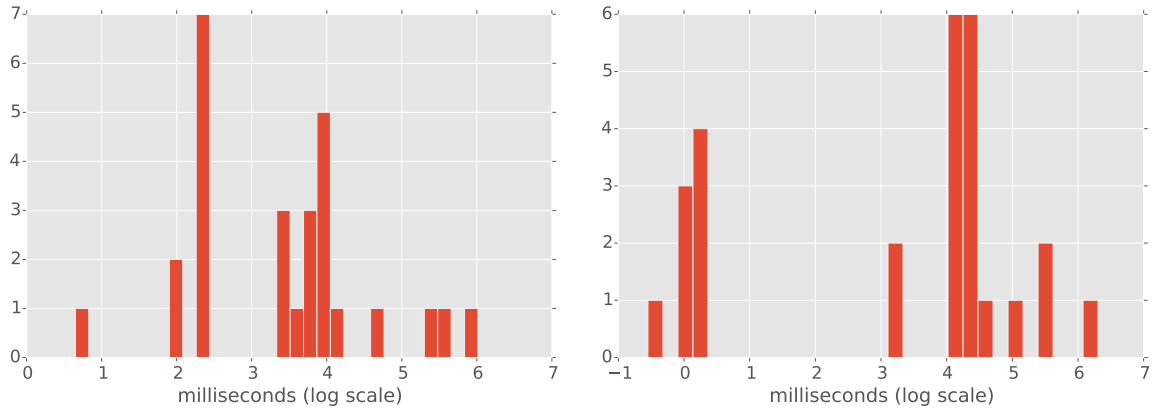


Figure 2.5: Histograms of the means (on the left) and standard deviations (on the right) of the difference between two successive timestamps of the 27 frames remaining in the *All modes* scenario after the dimension reduction. The ten rays with the lowest timestamps difference are the same on both histograms: only 10 frames have a regular sampling from 5 milliseconds to 200 milliseconds, whereas the 17 other frames have a low sampling frequency with a high variance.

The way we deal with the missing value depends on the sampling of each frame; the Figure 2.5 illustrates the diversity of the samplings by showing the mean and the standard deviation of

the timestamps difference of the frames in an acquisition file produced during an *All mode* scenario. Remember from section 2.1.2.2 that there are three kinds of sampling: periodic, contextual and irregular sampling.

Periodic sampling For the frame who have a regular sampling, we decide to fill the missing values by copying their last known values. From a radar perspective, that filling process is very coherent, because without a new emission, the radar is also keeping the last known value as the current one.

Contextual sampling The frames with a contextual sampling have the properties to be only categorical and to be emitted only when a specific event occurs, which means that without such an event, the frame may never be used: there is no value that corresponds to the *nothing to report* event. In agreement with the expert, we have defined *zero* as the default value for that *nothing to report* event. We have also defined a duration of 200 milliseconds for every event that these frames could report, before returning to *zero*. The missing values will thus be set to *zero* outside of the 200 milliseconds period following an emission, and copied inside that time interval.

Irregular sampling This class of sampling regroups the remaining frames which have a very chaotic behaviour. Just as in the case of the regular sampling, we decided to copy the last known value, since it's also the way the radar treats those frames.

Completion

Resampling

Completing the missing value with the last known ones results in a subsampling of the data: the high frequencies of the spectrograms of the variables will be cut. To minimise the effect of that completion, we have *resampled* the data to the frequency of the most emitted variable, which is 200 Hz: instead of creating a line for every emitted frame, we create a new line every 5 milliseconds, filled with the values received during the last 5 milliseconds. The missing values are yet still copied from the last known values.

These two data processings – completion and resampling – are producing the data file we will use in the following for the anomaly detection and localisation algorithms.

Example 2.6 Data processing of the *All Modes* and *Touch and Go* scenario
On the *all modes* scenario, this resampling process has produced a file of 839 columns and 2 100 000 lines. The 1004 columns are composed of 955 binary variables and 49 quantitative variables. With the *Touch and go* scenario, we ended up with a file of 86 columns and 1 600 000 lines, which are samples of 77 binary variables and 9 quantitative variables.

2.4 Anomaly detection and localisation

In this section, we define the anomaly detection and localisation problem.

2.4.1 Related works in anomaly detection

Anomaly detection refers to the task of detecting samples or patterns in data that do not behave according to a normal behaviour.

We can group the different kinds of anomalies into three groups (Chandola et al. [2009]):

Point anomalies: These anomalies are single samples which are anomalous with regard to the rest of the dataset. This group of anomalies has attracted most of the studies on anomaly detection.

Contextual anomalies: These anomalies relate to samples who are anomalous in a given context, but not otherwise. These anomalies are also termed as conditional anomalies (Valko et al. [2008], Valko [2011], Song et al. [2007]).

Collective anomalies: These anomalies refer to sets of samples that are anomalous given the rest of the data. The individual samples may not be anomalies when taken alone, but their succession or their collection is anomalous.

As explained in section 2.1.1, the built-in test if more focused on collective anomalies, since it is counting the emission of certain frames – which might not be anomalies if taken alone – and detect a breakdown when too much of those frames are received. On the other hand, contextual anomalies are very hard to detect, because the number of all possible contexts is too high. These contextual anomalies are the one we will be trying to detect in the following.

Anomaly detection has already attracted a lot of studies for many years (see Chandola et al. [2009], Hodge and Austin [2004], Agyemang et al. [2006] or Patcha and Park [2007] for exhaustive surveys on the topic), and has been applied in a large variety of applications: intrusion detection, fraud detection, medical and public health anomaly detection, industrial damage detection (also called health management), image processing, analyse of text data, etc.

The problems coming from the medical health anomaly detection field and the health management field are very close from our study. The data is typically records of various features of patients as the age, blood pressure and sugar level for the medical field, or sensor data for the industrial field, with a temporal aspect. Most of the study on that topic aim at detecting point anomalies – e.g., to detect anomalous records as well as instrumentation or recording errors – and collective anomalies – e.g., to detect anomalies in electroencephalograms or electrocardiograms.

Many authors has addressed the point detection anomaly problem and developed specific approaches, including parametric statistical modelling (Horn et al. [2001], Laurikkala et al. [2000], Guttormsson et al. [1999]), neural networks (Sakurada and Yairi [2014], Bennett and Campbell [2001]), Bayesian networks (Wong et al. [2003]), rule-based systems (Aggarwal [2005]), nearest neighbours based techniques (Lin et al. [2005]) and kernel-based weighted nearest neighbours techniques (Valizadegan and Tan [2007]), among many others.

The problem of detecting conditional anomalies has also attracted a lot of studies. The concept has been introduced by Dubitzky et al. [2007], where such anomalies where detected using Bayesian belief networks or naïve Bayes models , though this method did not scale to more than a dozen features. Valko et al. [2008] uses distances between hyperplanes learned by SVM to detect conditional anomalies in the medical field. Song et al. [2007] proposes a method where the users defines a partitioning of the features in two subsets, the indicator features that will directly be indicative for the anomaly, and the environmental features, which are not directly relevant for labelling but influence the indicator features. Valko [2011] proposes an approach based on nonparametric graph-based methods, relying on graph connectivity analysis and soft harmonic solution (Valko et al. [2011]).

In a detection problem, the data might have labels, which are often labeled as *anomalous* or *normal*. However, it is not always possible to have access to those labels or to have enough data from both categories. The possible approaches to solve an anomaly detection problem often depends on the available labels and the cost of their discovery. There are three main classes that distinguish all the anomaly detection techniques:

Unsupervised anomaly detection: techniques that operate in this mode are working on unlabeled data. The implicit assumption is made that the train data contains both normal samples and anomalous samples, where often the vast majority of the samples in the data is normal.

Supervised anomaly detection: techniques that operate in this mode are considering data where both the normal class and the anomaly class have samples. New unlabeled records are compared to a model to decide whether they should be classified as normal or anomalies. Often, it is very expensive to have a good representation of both the normal class and the anomaly class (often, the anomaly class is under-represented). In that sense, supervised anomaly detection is close to binary classification problems with very imbalanced classes.

Semi-supervised anomaly detection: techniques that operate in this mode make the assumptions that the training data – also called normal data – do not contain any anomaly and are drawn from the same unknown distribution. Finding a precise description of the normal data allows the detection of samples in new test sets that aren't drawn from this distribution.

In this thesis, we are facing a semi-supervised conditional anomaly detection problem that we will address using a graph-based parametric approach using probabilistic graphical models.

2.4.2 Graphical models for anomaly detection

Among the diversity of available techniques and approach to solve an anomaly detection problem, we will focus on the use of parametric statistical techniques based on graphical models.

Parametric statistical techniques make the assumption that the normal data have been generated from an unknown distribution p_Ω, where Ω is the set of parameters of the model. However, learning these parameters from data is often intractable, especially in high dimension. In the simplest case where all the n variables take 2 values, the joint distribution p_Ω has $2^n - 1$ parameters and its learning requires a lot of data, which, assuming these are available, possibly raises big training numerical issues. If we look at the data file produced by the scenario *Touch and go* (see example 2.6), the data is formed of instances of 69 variables, with 63 binary variables. If we consider the *All modes* scenario (see example 2.4), specifying a joint distribution of hundreds of variables appears totally intractable. More generally, manipulating such a big number of parameters is inconceivable from every perspective. On a computational point of view, it is too expensive to calculate those parameters, and too expensive to store. On a practical point of view, learning such a density from data would require having access to a huge number of samples, in order to have a good estimation of the underlying joint distribution. Those barriers are the main reasons to adopt graphical models.

Probabilistic Graphical Models

Probabilistic graphical models (Lauritzen [1996], Whittaker [2009] and Friedman and Koller [2009]) are a framework providing mechanisms for learning and exploiting the structure of complex distributions. It uses a graph-based representation to represent a complex distribution over a high-dimensional space. In this graphical representation, illustrated in Figure 2.6, the nodes correspond to the variables of our domain, and the edges correspond to direct probabilistic interactions between the variables.

A graphical model is an association between a graph $\mathcal{G}$, where the nodes of $\mathcal{G}$ are a set of random variables, and a distribution p_Ω of these random variables, where Ω is a set of parameters induced by the model family. The construction of graphical models is based on two equivalent perspectives: on one side, the graph is a representation of a set of conditional independences of random variables that hold in p_Ω, and on the other side, the graph is a skeleton over which the distribution p_Ω is factorised.

In the following, we will describe two families of graphical representation of distributions: the Bayesian networks – whose graphs are directed acyclic – and the Markov networks – whose graphs are undirected. For both families holds the duality of the two above perspectives, but they differ in the set of independences they can represent and in the way the distribution are factorised. For both Bayesian networks and Markov networks, we will describe how a model can be learned from data, and how that learned model can be used for detecting and localising anomalies.

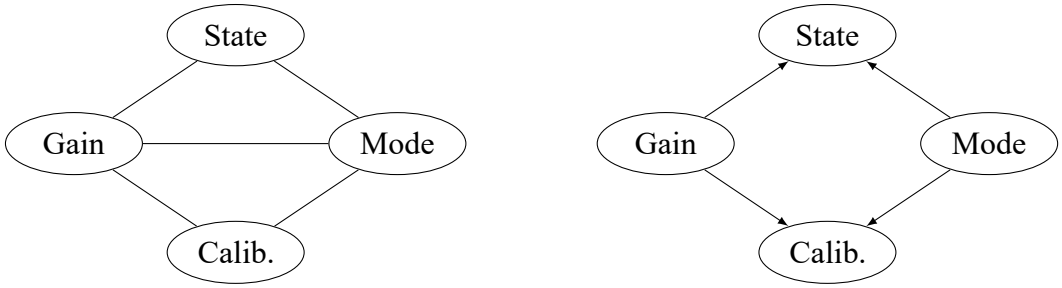


Figure 2.6: Two different perspectives on graphical models: undirected networks (on the left), also called Bayesian networks and directed networks (on the right), also called Markov networks.

2.4.3 Anomaly localisation

In the industrial field, localising an anomaly means to find the component(s) that caused the detected anomalies. Now that we have introduced the probabilistic graphical models framework, we can reformulate the localisation problem.

Suppose we have access to a distribution p_Ω from a graphical model – either learned from data or constructed by an expert team. We will be looking for anomalies that are located in the conditional distribution of each variable:

Definition 2.1 Localisation problem Suppose we have learned a model p_Ω with the parameters Ω over a set of n variables $x_1, \dots, x_n$, and we have a test set $\mathcal{D} = \{X^{(t)}, t = 1 \dots M\}$ of samples, indexed by the time. We define the localisation problem as finding the subset of variables $\{x_i, i \in 1, \dots, n\}$ whose conditional distributions $p(X_i^{(t)}|X_{-i}^{(t)})$, where x_{-i} denotes all the variables except x_i, monitored as a function of time, have changed compared to the learned model $p_\Omega(X_i^{(t)}|X_{-i}^{(t)})$.

As a first consequence of this definition, we see that the localisation task will be done alongside the detection task. The detection task thus relates to the detection of anomalous samples, whereas the localisation task aims at finding which variable(s) is (are) the cause of the anomalies. We can also remark that this definition does not depend on the class of graphical model, and both Bayesian networks and Markov networks can be used to achieve this goal, as long as the conditional probability distributions can be computed. In the next two sections, we will present these two model families, their learning process and how to use them for anomaly detection and localisation.

2.4.4 Bayesian networks

Bayesian Networks (Pearl [2014], Friedman and Koller [2009], on the right on Figure 2.6) are one way to model the joint distribution $p(x_1, \dots, x_n)$ of a set of n random variables. To avoid the intractability of the representation of the joint distribution, Bayesian networks exploit the

conditional independence properties holding in this joint distribution to provide a more compact representation.

2.4.4.1 Definition of a Bayesian Network

The representation of a Bayesian network is a Directed Acyclic Graph (DAG) $\mathcal{G}$, whose nodes are random variables of the domain and whose edges correspond to direct influences between variables. Since the edges are oriented, we can define the parents of a node x as the set of nodes $Pa_x^{\mathcal{G}}$ for which there is an oriented arrow $y \to x$ for $y \in Pa_x^{\mathcal{G}}$ in $\mathcal{G}$, and the children of a node x as the set of nodes y with $x \to y$ in $\mathcal{G}$.

A Bayesian network is often associated with a set of Conditional Probability Distributions (CPDs) that specify the distribution of the values of each variable given the values of every possible assignments of values of its parents in the graph $\mathcal{G}$. These CPDs are defining a joint probability distribution p via the *chain rule* for Bayesian networks

Chain rule for Bayesian networks

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | Pa_{x_i}^{\mathcal{G}}).$$

Factorisation

When a joint distribution can be expressed as the product of factors, where those factors are the conditional probabilities of each variable given its parent in the graph $\mathcal{G}$, we say that this distribution factorises over the graph $\mathcal{G}$.

Bayesian network

Definition 2.2 Bayesian Network A Bayesian network is a pair $\mathcal{B} = (\mathcal{G}, p_\Omega)$ where $\mathcal{G}$ is a directed acyclic graph, p_Ω is a distribution that factorises over $\mathcal{G}$, and where p_Ω is specified as a set of CPDs, parametrised by Ω, associated with the nodes of $\mathcal{G}$.

The common choices for the CPDs of a variable depend on the types of the variable and its parents:

- when both the variable and its parents are categorical, one can represent the CPD with a Conditional Probability Table (CPT,Heckerman et al. [1995]), which is a table providing the probability distribution of the variable for every assignment of values of its parents;
- when the variable and its parents are quantitative, the CPD is a conditional Gaussian distribution, whose parameters depend on the value of the parents (Geiger and Heckerman [1994]);
- when the variables and its parents are a mixture of quantitative and categorical variables, the CPD are either CPT or Gaussian distributions whose parameters depends on the value of continuous parents for each configuration of the categorical parents (Lauritzen and Wermuth [1989]).

To correctly take charge of Bayesian networks, we need to understand the relation between independences and factorisation. This requires the notion of d-separation and active trail in a graph. Let $\{i_1, \dots, i_k\}$ be a subset of k elements of $\{1, \dots, n\}$, $X_{i_1} \rightleftharpoons \cdots \rightleftharpoons X_{i_k}$ be a trail in $\mathcal{G}$ and $\boldsymbol{Z}$ be a subset of observed variables. We say that the trail $X_{i_1} \rightleftharpoons \cdots \rightleftharpoons X_{i_k}$ is *active* if:

Active trail

- whenever there is a structure $X_{i-1} \to X_i \leftarrow X_{i+1}$ (called v-structure), X_i or one of its descendant are not in $\boldsymbol{Z}$;
- there is no other node of the trail in $\boldsymbol{Z}$.

D-separation

This definition of an active trail leads to the definition of the *d-separation* (directed separation). Let $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ be three subset of nodes in $\mathcal{G}$, we say that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated given $\boldsymbol{Z}$ if there is no active trail between any nodes $X \in \boldsymbol{X}$ and $Y \in \boldsymbol{Y}$ given $\boldsymbol{Z}$. This notion of d-separation is the key to understand the relation between factorisation and independences:

Theorem 2.1 *For almost all distribution p that factorises over $\mathcal{G}$, i.e., for all distributions except for a set of measure zero in the space of CPD parametrisations, two subsets of variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated given $\boldsymbol{Z}$ if and only if the independence $(\boldsymbol{X} \perp \boldsymbol{Y} | \boldsymbol{Z})$ holds in p.*

This theorem stipulates that the independences of a distribution that factorises over a graph can be directly read from the graph, by finding the d-separated set of variables. For instance, the independences that will hold for each distribution that factorises over the Bayesian network structure in the Figure 2.6 (on the right) are $(G \perp M)$, $(S \perp C \mid G)$ and $(S \perp C \mid M)$.

In the next two sections, we discuss the problem of learning a Bayesian network using a dataset $\mathcal{D} = \{X^{(1)}, \dots, X^{(M)}\}$ consisting of M fully observed instances of the network variables. Learning a Bayesian network can be done in several ways, but it always decomposes in two steps: first learning the structure of the graph $\mathcal{G}$, and secondly learning the parameters Ω of the corresponding CPDs.

2.4.4.2 Learning the parameters of a Bayesian network

Suppose for now that the structure $\mathcal{G}$ of a Bayesian network is known, and that our training set $\mathcal{D} = \{X^{(1)}, \dots, X^{(M)}\}$ is composed of M i.i.d. drawn samples of the variables of the problem. We consider the parametric model $p_{\mathcal{G}}(X : \Omega)$, which is the family of distributions factorising over the structure $\mathcal{G}$ and parametrised by Ω, i.e., the distributions that will share CPDs with the same entries.

Likelihood function

Given the data $\mathcal{D}$, the *likelihood* function of the parameters Ω is

$$L(\Omega : \mathcal{D}) = \prod_{m=1}^{M} p_{\mathcal{G}}(X^{(m)} : \Omega).$$

Maximum Likelihood Estimator

The *Maximum Likelihood Estimator* (DeGroot et al. [1986], Bishop [2006]) is a method to choose the parameters $\hat{\Omega}$ given the dataset $\mathcal{D}$:

$$\hat{\Omega} = \underset{\Omega}{\operatorname{argmax}} L(\Omega : \mathcal{D}).$$

Disjoint parameters

We consider the case where the parameters Ω are *disjoint* , i.e., when the CPDs are parametrised by a separate set of parameters that are independent (Spiegelhalter and Lauritzen [1990]):

$$p_{\mathcal{G}}(X : \Omega) = \prod_{i=1}^{n} p_{\mathcal{G}}(X_i^{(m)} | Pa_{X_i}^{\mathcal{G}} : \Omega_{x|Pa_x^{\mathcal{G}}}),$$

where $\Omega_{x|Pa_x^{\mathcal{G}}}$ denotes the subset of parameters that determines $p_{\mathcal{G}}(X|Pa_X^{\mathcal{G}})$ in the model. In that case, the calculation of the likelihood is easily made due to the fact that it can be decomposed into a product of terms, one for each variable of the domain:

$$\begin{aligned} L(\Omega : \mathcal{D}) &= \prod_{m=1}^{M} p_{\mathcal{G}}(X^{(m)} : \Omega) \\ &= \prod_{i=1}^{n} \left[\prod_{m=1}^{M} p_{\mathcal{G}}(X_i^{(m)} | Pa_{X_i^{(m)}}^{\mathcal{G}} : \Omega_{X_i|Pa_{X_i}^{\mathcal{G}}}) \right]. \end{aligned}$$

This decomposition leads to the following property:

Proposition 2.2 (Global likelihood decomposition) *The likelihood function can be decomposed in a product of conditional likelihoods of each variable given its parents in $\mathcal{G}$:*

$$L(\Omega : \mathcal{D}) = \prod_{i=1}^{n} L_i(\Omega_{X_i|Pa_{X_i}^{\mathcal{G}}} : \mathcal{D}),$$

where $L_i(\Omega_{X_i|Pa_{X_i}^{\mathcal{G}}} : \mathcal{D})$ is the local likelihood function of the variable X_i.

This property leads to the main result for calculating the likelihood: we can maximise each local likelihood independently to find the MLE.

Proposition 2.3 *Let $\Omega_{X_i|Pa_{X_i}^{\mathcal{G}}}$ be the parameters of the CPD of X_i given its parents $Pa_{X_i}^{\mathcal{G}}$. Let $\hat{\Omega}_{X_i|Pa_{X_i}^{\mathcal{G}}}$ be the parameter that maximises the local likelihood $L_i(\Omega_{X_i|Pa_{X_i}^{\mathcal{G}}} : \mathcal{D})$. Then $\hat{\Omega} = \{\hat{\Omega}_{X_1|Pa_{X_1}^{\mathcal{G}}}, \ldots, \hat{\Omega}_{X_n|Pa_{X_n}^{\mathcal{G}}}\}$ maximises $L(\Omega : \mathcal{D})$.*

So far, we didn't make any assumption regarding the domain of the variables. In particular, the decomposition property apply to any type of CPD. The MLE approach is thus valid, whatever the domain of the variables. For a categorical variable X with categorical parents $\boldsymbol{U}$, we can

represent its CPD with a table, whose elements are $\Omega_{x|\boldsymbol{u}}$, where $x \in \text{Val}(X)$ and $\boldsymbol{u} \in \text{Val}(\boldsymbol{U})$. In that case, the local likelihood is given by

$$L_X(\boldsymbol{\Omega}_{X|\boldsymbol{U}}) = \prod_{\boldsymbol{u} \in \text{Val}(\boldsymbol{U})} \left[\prod_{x \in \text{Val}(X)} \Omega_{x|\boldsymbol{u}}^{M_{x|\boldsymbol{u}}} \right],$$

where $M_{x|\boldsymbol{u}}$ is the counter of the event $X = x$ and $\boldsymbol{U} = u$ in the dataset $\mathcal{D}$. We can then deduce the MLE parameters:

$$\hat{\Omega}_{x|\boldsymbol{u}} = \frac{M_{x|\boldsymbol{u}}}{M_{\boldsymbol{u}}},$$

where $M_{\boldsymbol{u}} = \sum_x M_{x|\boldsymbol{u}}$.

The case where some variables are not categorical leads to different formula for the local likelihoods, without changing the main approach. In the special case where all the variables are continuous (Geiger and Heckerman [1994]), the joint distribution of those variables is a multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma^{-1})$ with mean μ and covariance matrix Σ. This distribution can be rewritten as a product of conditional density

Gaussian Bayesian networks

$$p(x) = \prod_{i=1}^{n} p(x_i | x_{-i}),$$

each conditional density being an univariate Gaussian distribution

$$\mathcal{N}(\mu_i + \sum_{j \neq i} b_{ij}(x_j - \mu_j), \nu_i^{-1}),$$

where μ_i is the mean of x_i, ν_i is the conditional variance of x_i given x_{-i}, and b_{ij} is a coefficient indicating the strength of the interaction between x_i and x_j.

We can thus interpret a multivariate Gaussian distribution as a Bayesian network, where there is no edge from x_i to x_j when $b_{ij} = 0$. On the other hand, given a Bayesian network where the conditional probability distributions are all univariate Gaussian, we can reconstruct a multivariate Gaussian distribution. These networks are called Gaussian networks and have been studied by Shachter and Kenley [1989].

Conditional Gaussian Bayesian networks

The general case of networks having both categorical variables and quantitative variables are called *Conditional Gaussian Bayesian* networks and have been studied by Heckerman and Geiger [1995], Lauritzen and Wermuth [1989].

2.4.4.3 Learning the structure of a Bayesian network

In this section, we address the problem of learning the structure of a Bayesian network from data. We suppose the data $\mathcal{D}$ available for the learning are i.i.d. sampled from an implicit distribution $p^*_{\mathcal{G}}$, induced by an unknown Bayesian network $(\mathcal{G}, p^*_{\mathcal{G}})$.

First, we have to notice that the structure of the underlying Bayesian network is not identifiable from the data. Generally speaking, there are many network structures over which a distribution can be factorised. Remember from Theorem 2.1 that the structure of a Bayesian network encodes a set of independences. However, the same set of independences can be encoded by several structures, defining an equivalent class. To be convinced of this, one can consider the networks $A \leftarrow B$ and $A \rightarrow B$, which are encoding the same empty set of independences: over this networks are factorising the densities $P(A, B) = p(B)p(A|B)$ and $p(A, B) = P(A)P(B|A)$ which are strictly equivalent in term of independences.

The first thing to notice is that a network is hardly designable by human experts. It is admittedly feasible with a small number of variables, but becomes intractable as the dimension grows. There are mainly two approaches to learn a network structure ([Friedman and Koller, 2009, chapter 18]): constraint-based structure learning and score-based structure learning.

Constraint-based structure learning Constraint-based structure learning methods (Margaritis [2003], udea Pearl [1991]) are based on the learning of the set of independences of the distribution. They try to test every conditional dependences and independences in the data to find a member of the equivalent class of the network encoding the independences of the data. These methods are based on independence tests (Lehmann and Romano [2006]) that are subjects to failure, and will result in a wrong structure. Common tests are the mutual information test for categorical Bayesian networks, and the exact Student's T test for correlation for Gaussian Bayesian networks, introduced by Shachter and Kenley [1989] in the context of influence diagrams. However, testing every triplet $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$ to test if the conditional independences $\boldsymbol{X} \perp \boldsymbol{Y}|\boldsymbol{Z}$ holds for the data is intractable in hight dimension. Constraint-based method thus often limit the size of $\boldsymbol{Z}$ to one or two variables, what may also result in a wrong structure.

Score based structure learning Score-based approaches are considering the Bayesian structure learning problem as a model selection problem: they explore through a hypothesis space of candidate networks to find the highest-scoring structure. However, the space of Bayesian network structure is a combinatorial space containing a superexponential number of structures (2^{n^2} structures for n nodes). The exploration problem is thus $\mathcal{NP}$-hard, and require heuristics to explore the structure space. Most of the classic scores are based on the likelihood $L(\mathcal{G} : \mathcal{D})$ of the structure given the dataset, which decomposes as follow:

Proposition 2.4 (Likelihood score) *Let $\mathcal{D}$ be a dataset of M instances of the variables $X_1, \ldots, X_n$ of our domain. The likelihood score decomposes as*

$$L(\mathcal{G} : \mathcal{D}) = M \sum_{i=1}^{n} \mathbb{I}_{\hat{p}}(X_i, \mathrm{Pa}_{X_i}^{\mathcal{G}}) - M \sum_{i=1}^{n} \mathbb{H}_{\hat{p}}(X_i),$$

where $\hat{p}$ is the empirical distribution observed in the data $\mathcal{D}$, $\mathbb{I}_{\hat{p}}(\boldsymbol{X}, \boldsymbol{Y})$ is the mutual information between $\boldsymbol{X}$ and $\boldsymbol{Y}$ in the distribution $\hat{p}$ and $\mathbb{H}_{\hat{p}}(X)$ is the entropy of the variable X in $\hat{p}$.

Mutual Information

Remember that the *mutual information* quantifies how much a set of variables is informative about another set. For an arbitrary distribution p and some set of variables X and Y, the mutual information between X and Y in p is given by

$$\mathbb{I}_p(X, Y) = \sum_{x \in \mathrm{Val}(X), y \in \mathrm{Val}(Y)} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}.$$

Entropy

The *entropy* is a measure of the amount of uncertainty in a distribution: a low entropy implies that most of the distribution mass is spread on a few instances, whereas a high entropy implies a more uniform distribution. For an arbitrary distribution p and a set of variables X, the entropy of X in p is given by

$$\mathbb{H}_p(X) = \sum_{x \in \mathrm{Val}(X)} p(x) \log p(x).$$

Note that the entropy term does not discriminate the structures and can be removed without changing the result of the score maximisation. The likelihood score thus boils down to the calculation of mutual information, which can be interpreted as the strength of the dependences between the variables. That score will favour networks where the parents of each variable are informative about it. However, we can observe a property of the mutual information: for any $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$ and any distribution p,

$$\mathbb{I}_p(\boldsymbol{X}, \boldsymbol{Y} \cup \boldsymbol{Z}) \geq \mathbb{I}_p(\boldsymbol{X}, \boldsymbol{Y}).$$

Thus, a conditional independence will be exhibited by the maximum likelihood structure only when this independence holds exactly in the empirical distribution, which will practically never happen, especially when there is a lot of data. The structure learned by maximising the likelihood score will consequently be fully connected, which means that this score will overfit the data.

Overfitting

Several scores have been proposed to overcome this issue. Many authors have studied the impact of using priors $p(\mathcal{G})$ on the structure space to penalise certain structures or priors $p(\Omega|\mathcal{G})$ over the parameters values for a network $\mathcal{G}$. Using a Dirichlet prior for the parameters leads to

Bayesian Information Criterion

the classic *Bayesian Information Criterion* (Buntine [1991], Cooper and Herskovits [1992]):

$$\text{score}_{BIC}(\mathcal{G} : \mathcal{D}) = \log L(\mathcal{G} : \mathcal{D}) - \frac{\log M}{2}\text{Dim}(\mathcal{G}),$$

where $\text{Dim}(\mathcal{G})$ is the number of independent parameters in $\mathcal{G}$. This score seems to favour simple structures, and as it get more data, it will be able to recognize more complex structures.

The maximisation of any score though still requires exploring the structure space, which is a finite space. Chickering et al. [1995] compares different search algorithms, including K_2, local search or simulated annealing. Glover and Laguna [2013] studied the use of Tabu search.

2.4.4.4 Detecting and localising anomalies using Bayesian networks

Bayesian networks constitute a widespread class of graphical models to achieve anomaly detection, see Rashidi et al. [2011], Ye and Xu [2000], Wong et al. [2003], Lerner et al. [2000] and the references therein. Namely, given a Bayesian network $(\mathcal{G}, p_\Omega)$ whose parameters Ω of the conditional distributions are estimated from normal data, the computation of the likelihood is easily performed for new records of data to classify whether a record is anomalous or not. Indeed, the lower the likelihood, the higher the probability of having an anomalous record. This method has been successfully applied for network intrusion detection Ye and Xu [2000] and in the medical fields for disease outbreak detection Wong et al. [2003].

Once anomalous samples have been detected, the localisation task (see definition 2.1) can be done by analysing the conditional likelihood of each variable given its parents in the graph. The variables with anomalous values given the values of their parents have a higher probability of being the origin of the anomaly.

That approach has been successfully applied in Thales for detecting and localising anomalies in the data produced by the built-in test of the RDY radar, see Kemkemian et al. [2013]. In this study, a Bayesian network is learned over data, that are instances of 140 significant variables, using a commercial software called BayesiaLab [2013]. The learning was made by optimising a Minimum Description Length-like score, but the localisation inside low-likelihood samples had still to be manually done.

We present in section 5 an application of anomaly detection and localisation using Bayesian networks on true data from an RBE2 radar.

2.4.5 Undirected graphical models via the exponential family

We focus now on Markov networks, which is one of the two main graphical model families, alongside the Bayesian networks. One of the advantages of working with Markov networks is that we have no acyclicity constraint over the structure. This property will have many conse-

quences for the learning, and in particular, the likelihood can be optimised using classic convex optimisation algorithms – instead of local search heuristics – and has a unique optimum. However, the disadvantage of undirected models is that the log-likelihood does not decompose into a product of local likelihood as it was the case for Bayesian networks, and this makes parameter estimation much more expensive.

2.4.5.1 Definition of a Markov network

The representation of a Markov network is an undirected graph $\mathcal{G}$, whose nodes represent random variables and whose edges correspond to direct influences between variables. A Markov network is associated to a distribution p_Ω, which will have a different parametrisation and factorisation than the Bayesian networks. Namely, those distributions, called Gibbs distributions, are written as a product of *potentials* (or factors), which are simply positive functions over subsets of variables:

Potentials

$$p_\Omega(x_1, \dots, x_n) = \frac{1}{Z} \cdot \varphi_1(\boldsymbol{D}_1) \cdot \varphi_2(\boldsymbol{D}_2) \cdot \dots \cdot \varphi_k(\boldsymbol{D}_k),$$

where $\boldsymbol{D}_j, j = 1 \dots k$ are subsets of $\{x_1, \dots, x_n\}$, φ_j are the potentials associated to the subsets $\boldsymbol{D}_j$ and Z is the normalisation constant, also called partition function. Note that the potentials are not densities and thus did not sum to one, hence the partition function is mandatory to ensure that p is a valid distribution.

Partition function

We want now to relate the parametrisation of a Gibbs distribution to the structure of a Markov network. An undirected structure can be decomposed into complete sub-graphs, i.e., graphs where each node is connected to all the other nodes. These complete sub-graphs are called *cliques* – see Figure 2.7 for an example of clique decomposition of a graph structure – and the notion of clique decomposition leads to the definition of the factorisation of a distribution over a Markov network structure.

Cliques

Definition 2.3 Markov network factorisation We say that a distribution p_Ω parametrised by a set of potentials $\Omega = \{\varphi_1(\boldsymbol{D}_1), \dots, \varphi_k(\boldsymbol{D}_k)\}$ factorises over a Markov structure $\mathcal{G}$ if each $\boldsymbol{D}_j$, $j = 1 \dots k$ is a clique of $\mathcal{G}$, i.e., a complete sub-graph of $\mathcal{G}$. The potentials that parametrised p_Ω are hence called clique potentials.

We can now give the definition of a Markov network.

Definition 2.4 Markov network A Markov network is a pair $\mathcal{H} = (\mathcal{G}, p_\Omega)$ where $\mathcal{G}$ is an undirected graph and p_Ω is a distribution parametrised by a set of clique potentials that factorises over $\mathcal{G}$.

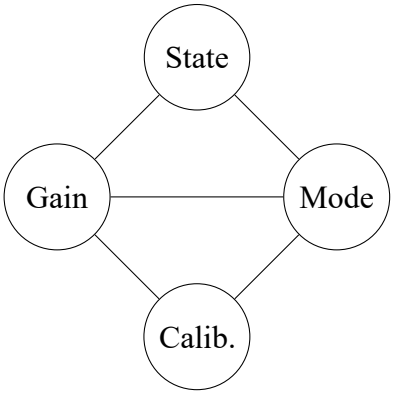


Figure 2.7: Markov network structure with four nodes State (S), Mode (M), Calibration (C) and Gain (G), corresponding respectively to four random variables S, M, C and G. This graph has six cliques, four formed by the pairs (S,M), (M,C), (C,G) and (G,S), and two formed by the triplets (S,G,M) and (G,M,C). We can also deduce the only independence holding for any distribution factorising over it, namely S $\perp$ C | (G, M).

Note that several distributions can factorise over a Markov network structure. For example, if we consider the network of the Figure 2.7, we can find at least two factorising distributions: one using maximal clique decomposition,

$$p(S, M, C, G) = \frac{1}{Z}\varphi_1(S, G, M)\varphi_2(C, G, M),$$

and another one using pairwise clique decomposition

$$p(S, M, C, G) = \frac{1}{Z}\varphi_1(S, G)\varphi_2(S, M)\varphi_3(M, C)\varphi_4(C, G)\varphi_5(M, G).$$

Separation

To fully understand the construction of Markov networks, we will describe how independences and factorisation are related. We define the notion of *separation*:

Definition 2.5 Separation We say that a set of nodes $\boldsymbol{Z}$ separates $\boldsymbol{X}$ and $\boldsymbol{Y}$ in a Markov network structure $\mathcal{G}$, if each path going from $X \in \boldsymbol{X}$ to $Y \in \boldsymbol{Y}$ contains at least a node from $\boldsymbol{Z}$.

As it was the case for Bayesian network (see Theorem 2.1), the notion of separation allows us to fully understand the relation between factorisation and independences.

Theorem 2.5 (Hammersley-Clifford) *For all distribution p that factorises over a structure $\mathcal{G}$, for any subsets of variables $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$, the conditional independence $(\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{Z})$ holds in p if and only if $\boldsymbol{Z}$ separates $\boldsymbol{X}$ and $\boldsymbol{Y}$ in $\mathcal{G}$.*

The Figure 2.7 illustrates how the structure of a Markov network can be used to discover any independences holding for a distribution that factorises over it.

We only postulate that the clique potentials are nothing more than positive functions: this allow us to provide a different representation for a Gibbs density. More precisely, we can rewrite any potential $\varphi(\boldsymbol{D})$ as

$$\varphi(\boldsymbol{D}) = \exp(-\epsilon(\boldsymbol{D})),$$

Energy function

where $\epsilon(\boldsymbol{D}) = -\log\varphi(\boldsymbol{D})$ is called an *energy function* – the use of that word comes from the statistical physics, where the probability of a physical state depends inversely on its energy. Usually, we reformulate the energy function as a weighted feature

$$\epsilon(\boldsymbol{D}) = -\omega_{\boldsymbol{D}} f_{\boldsymbol{D}}(\boldsymbol{D}),$$

where $\omega_{\boldsymbol{D}}$ is a positive weight and $f_{\boldsymbol{D}}$ a feature function of the variables $\boldsymbol{D}$. This reformulation leads to a more general framework for working with Markov networks.

Definition 2.6 Log-linear graphical models A distribution p_Ω can be expressed as a log-linear model associated with a Markov network structure $\mathcal{G}$ if p_Ω is associated with a set of features $f_1(\boldsymbol{D}_1), \ldots, f_k(\boldsymbol{D}_k)$ – where each $\boldsymbol{D}_j$ is a clique of $\mathcal{G}$ – and a set of weights $\Omega = \omega_1, \ldots, \omega_k$ such that

$$p_\Omega(x_1, \ldots, x_n) = \frac{1}{Z_\Omega} \exp\left[-\sum_{j=1}^{k} \omega_j f_j(\boldsymbol{D_j})\right]. \tag{2.1}$$

Pairwise networks

A subclass of networks that arises in many situations is the class of log-linear *pairwise Markov networks*, where the clique potentials are functions of one or two variables. More precisely, a distribution factorising over a pairwise Markov networks is associated with a set of node potentials $\{\varphi_i(x_i) = \omega_i f_i(x_i), i = 1 \ldots n\}$ and a set of edge potentials $\{\varphi_{ij}(x_i, x_j) = \omega_{ij} f_{ij}(x_i, x_j), i \neq j\}$, where the f_i and the f_{ij} are the associated features and $\Omega = \{\omega_i\}_i \cup \{\omega_{ij}, i \neq j\}$ are the associated weights. Note that the weights associated to edges are symmetric, i.e., $\omega_{ij} = \omega_{ji}$, for all $i \neq j$. In this framework, the factorising density has the form

$$p_\Omega(x_1, \ldots, x_n) = \frac{1}{Z_\Omega} \exp\left[\sum_{i=1}^{n} \omega_i f_i(x_i) + \sum_{i<j} \omega_{ij} f_{ij}(x_i, x_j)\right], \tag{2.2}$$

where the partition function is defined as

$$Z_\Omega = \int_x \exp\left[\sum_{i=1}^{n} \omega_i f_i(x_i) + \sum_{i \neq j} \omega_{ij} f_{ij}(x_i, x_j)\right] dx.$$

Note that in this framework, two variables x_i and x_j are conditionally independent to all the others variables if and only if $\omega_{ij} = 0$ and $\omega_{ji} = 0$. Indeed, the conditional distribution of x_i

given the other variables is

$$p_\Omega(x_i|x_{-i}) \propto \exp\left(\omega_i f_i(x_i) + f_i(x_i)\sum_{j>i}\omega_{ij}f_j(x_j)\right).$$

We thus see that when ω_{ij} is null, the conditional distribution of x_i doest not depend on x_j.

Pairwise Markov networks are attractive because of their simplicity and because edges interactions are a special case that often arises in practice. Though it is an important restriction in term of parametrisation, since pairwise networks have only $\mathcal{O}(n^2)$ independent parameters whereas the space of Markov network parametrisations is a combinatorial space containing a superexponential number of parametrisations, the pairwise framework embraces a large variety of models and in particular the exponential family, where the conditional distributions of each node arise from the exponential family. This subclass contains many of the most classic pairwise models, like the Ising model, the Potts model, or the Gaussian model.

2.4.5.2 Ising model and Potts model

The Ising model (Ising [1925], Wainwright and Jordan [2008]) is one of the earliest studied pairwise Markov networks, and was used in statistical physics to model the energy of a physical system involving interacting atoms. Each atom of this system corresponds to a binary-valued variable x_i, with values in $\{-1, 1\}$, defining the atom's spin. In this model, the energy function associated with the edges is symmetric and take the form

$$\epsilon_{ij}(x_i, x_j) = -\omega_{ij}x_i x_j,$$

and the node energy functions take the form

$$\epsilon_i(x_i) = -\omega_i x_i.$$

In the rest of our study, we will use the state space $\{0, 1\}$ for the binary-valued variables. It is convenient for the calculations, and in particular, we have $x_i^2 = x_i$ for each variable. The joint probability distribution (2.2) take the form

$$p_\Omega(x_1, \ldots, x_n) = \frac{1}{Z_\Omega}\exp\left[\sum_{i=1}^{n}\omega_i x_i + \sum_{i\neq j}\omega_{ij}x_i x_j\right]. \tag{2.3}$$

The term $\sum_{i=1}^{n}\omega_i x_i$ defines the distribution when there is no interactions. The parameters $\{\omega_{ij}\}$ are often constrained to be positive, since it corresponds to networks with only collaborative interactions between the nodes.

The Potts model Potts [1953] is a model where all the variables $x_1, \dots, x_n$ are categorical with the same state space $\{1, \dots, m\}$, and its density is the same as the density (3.3) of the Ising model.

Both models impose no condition on the parameters $\{\omega_i\}$ and $\{\omega_{ij}\}$, since there is a finite number of configurations for the binary and categorical variables, and the distribution is always normalisable.

2.4.5.3 Gaussian model

In a Gaussian model (Lauritzen [1996], Malioutov et al. [2006]), the joint distribution of variables n quantitative random variables $x_1, \dots, x_n$ is modelled a multivariate Gaussian distribution with mean vector μ and positive definite symmetric covariance matrix Σ:

$$p(x) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right]. \tag{2.4}$$

If we expand the term under the exponential in (2.4), we see that this equation is a special case of the pairwise Markov network (2.2). In particular, in a Gaussian model $(\mathcal{G}, p_\Omega)$ where $\Omega = (\mu, \Sigma)$, the matrix $\Delta = \Sigma^{-1}$ is called the *precision matrix*, and there is an edge in the graph $\mathcal{G}$ between the node i and j if and only if $\Delta_{ij} \neq 0$.

Precision matrix

An interesting property of the Gaussian model is that the precision matrix and the graph structure are related (Speed and Kiiveri [1986]). If two subsets of variables x_a and x_b are conditionally independent, then there is no edge between the corresponding nodes in the graph, and the corresponding block Δ_{ab} of the precision matrix is null. On the contrary, a null element in the covariance matrix Σ only states a marginal independence between the corresponding variables.

2.4.5.4 Exponential family

Often, when taken alone, the behaviour of variables can be easily represented by an appropriate model, like a Poisson distribution for count-valued variables, a Gaussian distribution for physical measures, or Bernoulli distribution for binary-valued variables. The generalisation from univariate distributions to a multivariate distribution has been done when all the univariate distributions are members of the exponential family, see Yang et al. [2015] in the general case and Lee and Hastie [2015], Laby et al. [2015] for mixing Bernoulli and Gaussian univariate conditional density.

Given a set of variables, we consider the case where ever every conditional probability dis-

Exponential family

tribution of each variable given all the others is a member of the *exponential family*:

$$p(x_i|x_{-i}) = \frac{1}{Z_i(x_{-i})} \exp[E_i(x_{-i})B_i(x_i) + C_i(x_i)], \tag{2.5}$$

where x_{-i} represent all the variables except x_i. The sufficient statistic $B_i(\cdot)$ and the base measure $C_i(\cdot)$ are specified by the choice of the univariate exponential family, the function $E_i(\cdot)$ is an arbitrary function of x_{-i}, and $Z_i(\cdot)$ is the normalisation constant. Then these conditional distributions are consistent with a joint distribution p_Ω factoring over a graph $\mathcal{G}$, i.e., a Markov network $(\mathcal{G}, p_\Omega)$ with cliques of size at most K, if for each $i = 1 \dots n$, the functions $E_i(\cdot)$ have the following form (Yang et al. [2014]):

$$\omega_i + \sum_{j=1}^{n} \omega_{ij} B_j(x_i) + \cdots + \sum_{j_1,\dots,j_K=1}^{n} \omega_{ij_1\dots j_K} \prod_{k=2}^{K} B_{j_k}(x_{t_k}).$$

In that case, the joint distribution take the form

$$p_\Omega(x_1, \dots, x_n) = \frac{1}{Z_\Omega} \exp\left[\sum_{i=1}^{n} \omega_i B_i(x_i) + \cdots + \sum_{i_1,\dots,i_K=1}^{n} \omega_{i_1,\dots,i_K} \prod_{k=1}^{K} B_{i_k}(x_{i_k}) + \sum_{i=1}^{n} C_i(x_i)\right],$$

where $\Omega = \{\omega_1, \dots, \omega_{n,\dots,n}\}$. In the special case of pairwise models, i.e., when $K = 2$, the density above takes the form

$$p_\Omega(x_1, \dots, x_n) = \frac{1}{Z_\Omega} \exp\left[\sum_{i=1}^{n} \omega_i B_i(x_i) + \sum_{i,j=1}^{n} \omega_{ij} B_i(x_i) B_j(x_j) + \sum_{i=1}^{n} C_i(x_i)\right], \tag{2.6}$$

where Z_Ω is the partition function

$$Z_\Omega = \int_x \exp\left[\sum_{i=1}^{n} \omega_i B_i(x_i) + \sum_{i,j=1}^{n} \omega_{ij} B_i(x_i) B_j(x_j) + \sum_{i=1}^{n} C_i(x_i)\right] dx.$$

Note that the partition function Z_Ω might not be always integrable and its integrability has to be checked for each different mixing.

Manichean graphical models

The case where the variables can be regrouped in one or two types is called *manichean graphical models* (Yang et al. [2014]). Under this framework, the variables can be partitioned in two groups, $\boldsymbol{Y}$ taking values in $\mathcal{Y}$, and $\boldsymbol{Z}$ taking values in $\mathcal{Z}$. Depending on the cardinal of $\mathcal{Y}$ and $\mathcal{Z}$, it is possible to know if $\boldsymbol{Y}$ and $\boldsymbol{Z}$ can be mixed.

If both $\mathcal{Y}$ and $\mathcal{Z}$ are finite, then the joint distribution over $\boldsymbol{Y}$ and $\boldsymbol{Z}$ is normalisable, since it requires the summation of a finite number of cases.

If either $\mathcal{Y}$ or $\mathcal{Z}$ is finite, say $\mathcal{Y}$ without lost of generality, then the following theorem applies (Yang et al. [2014]):

Theorem 2.6 *If the domain $\mathcal{Y}$ is finite with $\max\{\mathcal{Y}\} < +\infty$ and $\min\{y \in \mathcal{Y}\} > -\infty$, and if the conditional distribution $p(\boldsymbol{Z}|\boldsymbol{Y})$ is normalisable for all $\boldsymbol{Y} \in \mathcal{Y}$, then the partition function is finite and the pairwise joint distribution $p_\Omega(\mathcal{Y}, \mathcal{Z})$ is normalisable.*

In particular, this theorem shows that mixing Gaussian (where $B_i(x_i) = \frac{x_i}{\sigma_i}$ and $C_i(x_i) = -\frac{x_i^2}{2\sigma_i^2}$) and Ising graphical models (where $B_i(x_i) = x_i$ and $C_i(x_i) = 0$) leads to a valid mixed joint distributions. The domain of the binary variables of the Ising model is $\{0, 1\}$ and is finite, and the conditional distribution of the Gaussian variables given the binary one's is well defined (see Lee and Hastie [2015], Laby et al. [2015] and the section 3 of this thesis). However, mixing Poisson (where $B_i(x_i) = x_i$ and $C_i(x_i) = -\log(x_i!)$) and Ising graphical models does lead to a valid mixed joint distribution only if $\omega_{ij} \leq 0$ for all i, j corresponding to the Poisson variables (Yang et al. [2013]).

If both $\mathcal{Y}$ and $\mathcal{Z}$ are infinite, with $\sup\{y \in \mathcal{Y}\} = \infty$ or $\inf\{y \in \mathcal{Y}\} = -\infty$ with the same for $\mathcal{Z}$, for all class distributions with linear sufficient statistic $B_i(x_i) = x_i$ (which include the popular distributions like Poisson, Gaussian, Bernoulli, Ising, exponential, ...), the following theorem defines the valid mixed distributions:

Theorem 2.7 (Yang et al. [2014]) *If both the domains $\mathcal{Y}$ and $\mathcal{Z}$ are infinite, if the sufficient statistic for each variable is the identity function, then the mixed joint distribution (2.2) is not normalisable if neither of the following conditions holds, for all i, j with $\omega_{ij} \neq 0$:*

1. *the domain of x_i and x_j are both infinite only from one side,*
2. *for all $\alpha, \beta > 0$ such that $-C_i(x_i) = \mathcal{O}(X_i^\alpha)$ and $-C_j(x_j) = \mathcal{O}(X_j^\beta)$, we have $(\alpha - 1)(\beta - 1) \geq 1$.*

In particular, the Gaussian - Poisson mixed distribution is not a valid one. Without loss of generality, suppose that the conditional distribution of the variables $\boldsymbol{Y}$ are univariate Gaussian distribution with known variance σ^2, and the variables $\boldsymbol{Z}$ correspond to a Poisson distribution. Since the domain $\mathcal{Y}$ of the Gaussian variables is $\mathbb{R}$, it is infinite in both directions and the first condition 1. of the theorem 2.7 is thus not satisfied. Concerning the second condition, $\alpha = 2$ since $C_{\boldsymbol{Y}}(Y_i) = -\frac{Y_i^2}{\sigma^2}$. Moreover, $\log(z_i!) \sim z_i \log(z_i)$ so $-C_{\boldsymbol{Z}}(z_i) = \mathcal{O}(z_i^\beta)$ for any $\beta > 1$. The second condition is also not valid, and mixing Poisson and Gaussian distribution is only possible if there is no interaction between $\boldsymbol{Y}$ and $\boldsymbol{Y}$.

2.4.5.5 Model learning

The learning of a Markov network from data is a quite different task than the learning of a Bayesian networks. If the problem often also relates to the calculation of the maximum likelihood estimator, there is some main differences in the learning of both kinds of model:

1. first, there is no acyclicity constraint for the Markov networks, which allows the use of convex optimisation techniques for optimising the likelihood. The use of ℓ_1 regularisation for learning sparse structure has been extensively studied, see Schmidt [2010] for an exhaustive survey.

2. Secondly, under the log-linear framework, the likelihood function is a strictly concave function of its parameters, which guarantees the existence of a unique optimum.

3. Finally, where the joint distribution of Bayesian networks was a product of conditional probability distributions, the joint distribution of a Markov network has a global normalisation constant Z that prevents the decomposition of the learning into local parameters learning problem, as it was the case for Bayesian networks. This global parameter has significant computational consequences, because in many cases, it has no closed-form and makes the exact calculation of the maximum likelihood estimator unachievable in practice.

Except for the case of the Gaussian model where the maximum likelihood estimator has a known closed-formed, learning a graphical model, especially in the case of mixed distributions, is still a challenging issue. The intractability of the calculation of the partition function forces the use of iterative methods, for optimizing over the parameter space. The next chapter will address the learning problem of a Markov network.

2.5 Motivation for the study

We chose to address the anomaly detection and localisation problem of Thales with a semi-supervised fashion by using probabilistic graphical models. The benefits of using Bayesian networks had already been shown by Kemkemian et al. [2013], nevertheless we decided to investigate a different direction, because of the lack of guarantee that offers the Bayesian network learning. Namely, the structure space of Bayesian networks is a combinatorial space, and the underlying structure is not identifiable from the data.

In contrast, the Markov network framework provides the theoretical guarantees that will make their learning much more reliable: the hypothesis state is continuous, and the likelihood has a unique optimum, what allows the use of convex optimisation algorithm and ℓ_1 regularisation for learning sparse structures.

The data provided by Thales are instances of heterogeneous variables, and in the following chapters, we will describe how to learn a mixed pairwise Markov network to model the normal behaviour of a radar and how to exploit this learned model to detect and locate anomalies in new records.

Chapter 3 addresses the learning of a mixed network, with two approaches: the first uses a stochastic version of the proximal gradient algorithm (Atchade et al. [2015]) that approximates the partition function Z_Ω through MCMC simulations, and the second uses the pseudo-likelihood (Besag [1975]) instead of the classic likelihood to avoid the intractability of the calculation of the partition function.

Chapter 4 presents our anomaly detection and localisation method. This technique is based on the monitoring of the conditional probability distributions as a function of time, and aim at detecting a change in this conditional distributions by using a two-sided version of the CUSUM algorithm (Basseville et al. [1993], Page [1954]).

In the chapter 5 we thoroughly investigate how the proposed methods performs on real data coming from the RBE2 radar production in Thales and provide insights about the practical application and results of our methods.

Chapter 3

Learning a mixed undirected graphical model

In this chapter, we will present the mixed graphical model framework and how to learn a mixed graphical model from data. The data we are considering are instances of heterogeneous variables, some being quantitative like gains, phases, temperatures, and some being categorical like working states or modes. In the following sections, we will denote by X an instance of the variables x, with $x = (x_{\mathcal{C}}, x_{\mathcal{Q}})$ where $x_{\mathcal{C}} = (x_i, i \in \mathcal{C})$ are the categorical variables where and $x_{\mathcal{Q}} = (x_u, u \in \mathcal{Q})$ are the quantitative variables. Here $\mathcal{C}$ and $\mathcal{Q}$ respectively refer to the indices of categorical and quantitative variables of x.

Many authors have addressed the graphical model learning problem in the case where the data is either categorical or quantitative, but only a few works are considering the heterogeneous case. Though this field is too vast for exhaustively enumerating all the studies, we propose a review of related works on that topic in the section 3.2.2.

We present in section 3.1 the mixed graphical model framework we designed for the anomaly detection and localisation issue of Thales. In this model, conditionally to the other variables, each variable will be associated either with a Bernoulli distribution if the variable is categorical or with a univariate Gaussian density if the variable is quantitative. In section 3.2, we then propose two algorithms we developed to learn a mixed graphical model from data. In section 3.3, we compare our learning algorithms with other standard techniques on synthetic data.

3.1 Mixed model presentation

In this section, we present the mixed model framework that we will be using throughout this thesis.

3.1.1 The mixed model framework

As explained in section 2.4.5, the pairwise undirected graphical model framework is attractive from many perspectives. The pairwise restriction is a good compromise between richness of the model and learning complexity. On a computational point of view, the complexity of a learning algorithm in the pairwise framework won't scale exponentially with the dimension, what makes them tractable in high dimension. For example, a pairwise network with n nodes associated to n variables has maximum $\begin{pmatrix} n \\ 2 \end{pmatrix} = n(n-1)/2$ edges potentials, whereas a model with no restrictions on the clique's sizes can have 2^n edges potentials. Of course, when n is large, the dimension of the model still increases significantly. This will be taken into account in the following. Another nice feature of the model is that the likelihood function is concave over a continuous space of parametrisation, which allows us to rely on well understood numerical procedures for statistical inference.

The model we propose is a pairwise undirected graphical model mixing a Gaussian model, used with continuous variables with values in $\mathbb{R}$, and an Ising model, used with binary variables with values in $\{0,1\}$ or $\{-1,1\}$. The Gaussian model is parametrised by a positive definite covariance matrix Σ and a mean vector μ, and its density is given by

$$p_{\Sigma,\mu}(x) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right],$$

which can be formulated using the precision matrix $\Delta = \Sigma^{-1}$, in which case the density takes the form

$$p_{\Delta,\mu}(x) = \frac{1}{Z_{\Delta,\mu}} \exp\left[\mu^T\Delta x - \frac{1}{2}x^T\Delta x\right], \tag{3.1}$$

where here $Z_{\Delta,\mu}$ denotes the normalisation constant. On the other hand, the Ising model is parametrised by $\Theta = \{(\theta_i)_{i=1\dots n}, (\theta_{ij})_{i>j})\}$, and its distribution is given by

$$p_\Theta(x) = \frac{1}{Z_\Theta} \exp\left[\sum_{i=1}\theta_i x_i + \sum_{i>j}\theta_{ij}x_i x_j\right], \tag{3.2}$$

where Z_Θ denotes the normalisation constant. In this work, we will use the state space $\{0,1\}$ for the binary variables x_i, hence $x_i = x_i^2$. The Ising distribution can be reexpressed as

$$p_\Theta(x) = \frac{1}{Z_\Theta} \exp\left[\sum_{i,j=1}^{n}\theta_{ij}x_i x_j\right] = \frac{1}{Z_\Theta}\exp(x^T\Theta x), \tag{3.3}$$

where we have kept the same notation Θ for the parameters of the model as in equation (3.2).

Note also that we are considering here categorical binary variables instead of general categorical variables, either binary or non-binary. It is possible to use the Potts model (2.3) for modelling non-binary categorical variables, however this model can be used only when all the variables have the same states space and when these states are comparable between variables. Though all categorical variables used by the built-in test take the same states – hexadecimal numbers between `0000` and `FFFF`, see section 2.1.2 – these states are not comparable between each others and have different meaning for different variables. To address this issue, we will binarise the non-binary categorical variables using the classic *1-to-K encoding scheme*, as proposed by [Bishop, 2006, §4.3.4] and Schmidt [2010]. The principle is the following: for $i \in \mathcal{C}$, if x_i takes values in $1, \dots, m_i$, we use instead the binary vector $t^{(i)} \in \{0,1\}^{m_i}$, with $t^{(i)}_{k_0} = 1$ if $x_i = k_0$, and $t^{(i)}_k = 0$ elsewhere for $k \neq k_0$. This transformation will only be done for non-binary categorical variables and thus will only impact Θ and Φ, whose dimensions will be consequently increased. Note that the binary variables are not concerned by this transformation. Thereafter in this paper, when we use the notation X, x, $X_{\mathcal{C}}$ and $x_{\mathcal{C}}$, we will suppose that the non-binary categorical data were already transformed following this scheme.

1-to-K encoding scheme

We can now define the pairwise undirected mixed graphical model framework.

Definition 3.1 Mixed pairwise graphical model For heterogeneous variables $x = (x_{\mathcal{C}}, x_{\mathcal{Q}})$ with $x_{\mathcal{C}} \in \{0,1\}^{|\mathcal{C}|}$ and $x_{\mathcal{Q}} \in \mathbb{R}^{|\mathcal{Q}|}$, we use the *pairwise undirected mixed graphical model*

Mixed graphical model

$$p_\Omega(x) = \frac{1}{Z_\Omega} \exp\left[x_{\mathcal{C}}^T \Theta x_{\mathcal{C}} + \mu^T x_{\mathcal{Q}} - \frac{1}{2} x_{\mathcal{Q}}^T \Delta x_{\mathcal{Q}} + x_{\mathcal{C}}^T \Phi x_{\mathcal{Q}}\right], \tag{3.4}$$

where $\Omega = (\Theta, \mu, \Delta, \Phi)$ contains all the parameters of the model, and where Z_Ω is the partition function defined by

$$Z_\Omega = \sum_{x_{\mathcal{C}} \in \{0,1\}^{|\mathcal{C}|}} \int_{\mathbb{R}^{|\mathcal{Q}|}} \exp\left[x_{\mathcal{C}}^T \Theta x_{\mathcal{C}} + \mu^T x_{\mathcal{Q}} - \frac{1}{2} x_{\mathcal{Q}}^T \Delta x_{\mathcal{Q}} + x_{\mathcal{C}}^T \Phi x_{\mathcal{Q}}\right] dx_{\mathcal{Q}}. \tag{3.5}$$

Here, $\Theta = (\theta_{ij})_{i,j \in \mathcal{C}}$ is a symmetric matrix, $\mu = (\mu_i)_{i \in \mathcal{Q}} \in \mathbb{R}^{\mathcal{Q}}$, $\Delta = (\delta_{uv})_{u,v \in \mathcal{Q}}$ is a positive definite symmetric matrix and $\Phi = (\phi_{iu})_{i,u \in \mathcal{C} \times \mathcal{Q}}$ is a general matrix. To simplify the notation, we have mixed the subset of variables indices $\mathcal{C}$ and $\mathcal{Q}$ with the indices of the corresponding matrices.

As explained earlier, the mixed model (3.4) involved a mix between an Ising model modelling the interactions between the categorical variables, a Gaussian model modelling the interactions between the quantitative variables and a last term for interactions between the categorical and the quantitative variables. In order for p_Ω to be a valid density with respect to the product measure made up of the counting measure on $\{0,1\}^{|\mathcal{C}|}$ and the Lebesgue measure on $\mathbb{R}^{|\mathcal{Q}|}$, one just requires

Δ to be a positive definite symmetric matrix. No other condition is imposed on μ, Θ and Φ other than Θ symmetric.

3.1.2 Properties of the mixed models

The mixed model (3.4) has been studied by Laby et al. [2015] and Lee and Hastie [2015], and is a special case of the joint density defined via exponential families introduced by Yang et al. [2015]. In particular, the theorem 2.6 shows that the mixed density (3.4) is valid if the conditional densities $p_\Omega(x_\mathcal{Q}|x_\mathcal{C})$ are normalisable.

Seen as a function of $x_\mathcal{Q}$ only, the density (3.4) can be rewritten as

$$p_\Omega(x_\mathcal{Q}|x_\mathcal{C}) \propto \exp\left(\left(\mu^T + x_\mathcal{C}^T\Phi\right) x_\mathcal{Q} - \frac{1}{2} x_\mathcal{Q}^T \Delta x_\mathcal{Q} \right),$$

where $\propto$ means equality between functions up to a multiplicative constant depending on $x_\mathcal{C}$. We recognise in the right part of the equation a special case of the Gaussian density (3.1). We can thus conclude that, given $x_\mathcal{C}$, $x_\mathcal{Q}$ has a Gaussian distribution with mean $\Delta^{-1}\left(\mu + \Phi^T x_\mathcal{C}\right)$ and covariance matrix Δ^{-1}.

A similar property holds for the conditional distribution of $x_\mathcal{C}$. Seen as a function of $x_\mathcal{Q}$, the density (3.4) becomes

$$\begin{aligned} p_\Omega(x_\mathcal{C}|x_\mathcal{Q}) &\propto \exp\left(x_\mathcal{C}^T\Theta x_\mathcal{C} + x_\mathcal{C}^T\Phi x_\mathcal{Q}\right) \\ &\propto \exp\left(x_\mathcal{C}^T\left[\Theta + \mathrm{Diag}\left(\Phi x_\mathcal{Q}\right)\right] x_\mathcal{C}\right), \end{aligned}$$

where $\mathrm{Diag}\,(u)$ is the diagonal matrix with the vector u on its diagonal, and where we have used $x_i^2 = x_i$ for $i \in \mathcal{C}$. We recognise the Ising model with parameter $\Theta + \mathrm{Diag}\,(\Phi x_\mathcal{Q})$.

More surprisingly, the marginal distribution of $x_\mathcal{C}$ is still an Ising model. Indeed, we have that

$$p_\Omega(x_\mathcal{C}) \propto \exp(x_\mathcal{C}^T\Theta x_\mathcal{C}) \int_{\mathbb{R}^{|\mathcal{Q}|}} \exp\left((\mu + \Phi^T x_\mathcal{C})^T x_\mathcal{Q} - \frac{1}{2} x_\mathcal{Q}^T \Delta x_\mathcal{Q} \right) dx_\mathcal{Q}.$$

We can actually interpret the integral term (up to a multiplicative constant) as the expectation $\mathbb{E}[\exp\left((\mu + \Phi^T x_\mathcal{C})^T U\right)]$ where U is a Gaussian vector with zero mean and covariance Δ^{-1}. We thus get

$$\begin{aligned} p_\Omega(x_\mathcal{C}) &\propto \exp\left(x_\mathcal{C}^T\Theta x_\mathcal{C} + \frac{1}{2}(\mu + \Phi^T x_\mathcal{C})^T \Delta^{-1} (\mu + \Phi^T x_\mathcal{C}) \right) \\ &\propto \exp\left(x_\mathcal{C}^T\left[\Theta + \Phi\Delta^{-1}\Phi^T/2 + \mathrm{Diag}\left(\Phi\Delta^{-1}\mu\right)\right] x_\mathcal{C}\right). \end{aligned}$$

where Diag (u) is the diagonal matrix with the vector u on its diagonal. We recognise an Ising model with parameter $\Theta + \Phi\Delta^{-1}\Phi^T/2 + \text{Diag}\,(\Phi\Delta^{-1}\mu)$.

However, there is no such similar property for the marginal distribution of $x_{\mathcal{Q}}$, which is a Gaussian distribution if and only if $\Phi = 0$, in which case this Gaussian distribution is parametrised by a mean vector $\Delta^{-1}\mu$ and a covariance matrix Δ^{-1}. In every other cases, $p_\Omega(x_{\mathcal{Q}})$ is a mixture of Gaussian distributions.

This four results are summarised in the following proposition.

Proposition 3.1 *We denote by $x_{\mathcal{C}}$ and $x_{\mathcal{Q}}$ respectively the binary and quantitative variables x, with $x = (x_{\mathcal{C}}, xq)$ where $x_{\mathcal{C}} \in \{0,1\}^{|\mathcal{C}|}$ and $x_{\mathcal{Q}} \in \mathbb{R}^{|\mathcal{Q}|}$. Then the four following properties are holding for any mixed model defined by the definition 3.1 and parametrised by $\Omega = (\Theta, \mu, \Delta, \Phi)$:*

i. Given $x_{\mathcal{C}}$, the conditional distribution of $x_{\mathcal{Q}}$ is Gaussian with mean $\Delta^{-1}\left(\mu + \Phi^T x_{\mathcal{C}}\right)$ and covariance matrix Δ^{-1}.

ii. Given $x_{\mathcal{Q}}$, the conditional distribution of $x_{\mathcal{C}}$ is a Ising model with parameters $\Theta + \text{Diag}\,(\Phi x_{\mathcal{Q}})$.

iii. The marginal distribution of $x_{\mathcal{C}}$ is an Ising model with parameters $\Theta + \Phi\Delta^{-1}\Phi^T/2 + \text{Diag}\,(\Phi\Delta^{-1}\mu)$.

iv. The marginal distribution of $x_{\mathcal{Q}}$ is a mixture of Gaussian distributions, except when $\Phi = 0$, in which case it is Gaussian with mean $\Delta^{-1}\mu$ and covariance Δ^{-1}.

Two of this properties are illustrated on the Figure 3.1, where we have shown some simulations of the mixed density (3.4) in the case $\Phi = 0$ (on the left) and in the case $\Phi \neq 0$ (on the right). The sampling process will be described later in the Algorithm 2. The property 3.1.i, stating that the quantitative variables have a conditional Gaussian distribution, is illustrated on both figures, where all samples with the same color have a Gaussian distribution. The property 3.1.iv, stating that the marginal density of $x_{\mathcal{Q}}$ is Gaussian only if $\Phi = 0$, is also illustrated on this plots: on the left where $\Phi = 0$, the quantitative samples are actually Gaussian distributed, whereas on the right where $\Phi \neq 0$, the quantitative samples doesn't have a multivariate Gaussian distribution but rather a mixture of Gaussian distributions.

3.1.3 From distribution to graphs

Let us explore more deeply the relation between mixed undirected graphs and mixed distributions. Remember that, as explained in section 2.4.5.1, in the framework of undirected models, the Hammersley-Clifford theorem 2.5 states that the absence of an edge between two nodes x_i and x_j states that x_i and x_j are conditionally independent given all the others variables x_{-ij}: observing x_{-ij} actually separates x_i and x_j, since there is no active chain between this two nodes

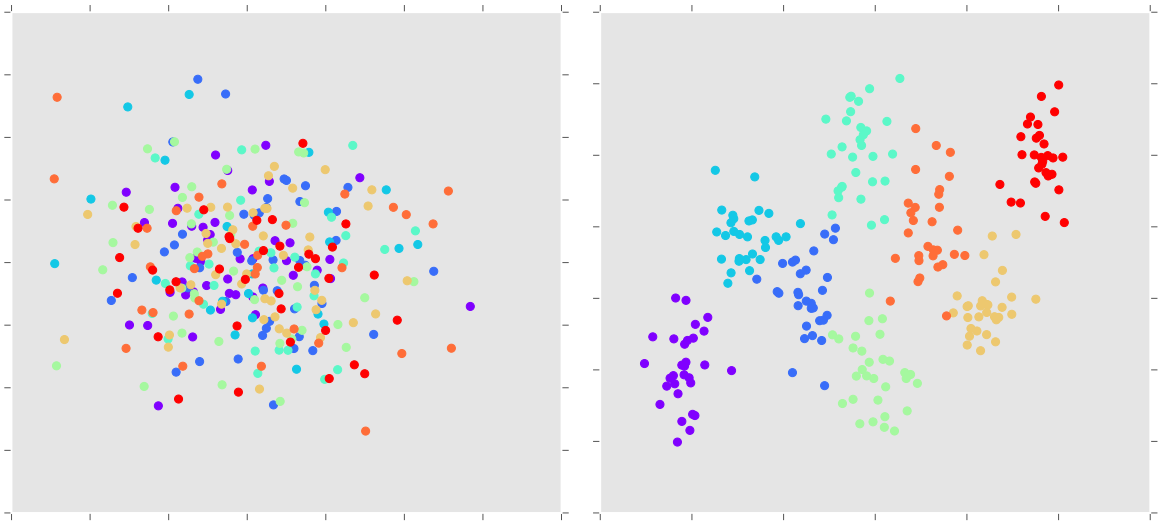


Figure 3.1: Illustration of i.i.d. samples of the mixed density (3.4) with 2 quantitative variables and 3 binary variables. On both figures, the binary variables are represented by $2^3 = 8$ colours, and the two quantitative variables are displayed along the two axis. The left figure illustrates the case $\Phi = 0$, i.e., where $x_{\mathcal{Q}}$ and $x_{\mathcal{C}}$ are independent, and the right figure illustrates the case $\Phi \neq 0$, i.e., where $x_{\mathcal{Q}}$ and $x_{\mathcal{C}}$ are dependent.

(see definition 2.5). It also implies that $p_\Omega(x_i|x_{-i})$ is independent of x_j, and $p_\Omega(x_j|x_{-j})$ is independent of x_i, where x_{-k} denotes all the variables except x_k. This conditional independence is thus equivalent of having a null weight for the edge potential of x_i and x_j.

The discussion about the relation between independences and edge potentials in section 2.4.5.1 also shows that the weights associated to the edges potentials, i.e., θ_{ij}, δ_{ij} or ϕ_{ij} depending on the domain of the variables x_i and x_j, only occur in the two conditional distributions $p_\Omega(x_i|x_{-i})$ and $p_\Omega(x_j|x_{-j})$. Let us compute the conditional distributions of the quantitative and categorical variables to see how these properties hold for the mixed model.

In the case where x_i is categorical, for $i \in \mathcal{C}$, we have that

$$p_\Omega(x_i|x_{-i}) \propto \exp\left[\theta_{ii}x_i^2 + x_i\left(2\sum_{j>i}\theta_{ij}x_j + \sum_{u\in\mathcal{Q}}\phi_{iu}x_u\right)\right], \tag{3.6}$$

where we have used x_u for the u-th entry of $X_{\mathcal{Q}}$ to keep the notation simple. We see that the only parameters involved in the conditional distribution of x_i given x_{-i} are $\{\theta_{ij}\}_{j\in\mathcal{C}}$ and $\{\phi_{iu}\}_{u\in\mathcal{Q}}$.

In the case where x_u is quantitative, for $u \in \mathcal{Q}$, we have that

$$p_\Omega(x_u|x_{-u}) \propto \exp\left[-\frac{1}{2}\Delta_{uu}x_u^2 + x_u\left(-\sum_{v>u}\Delta_{uv}x_v + \sum_{i\in\mathcal{C}}\phi_{iu}x_i\right)\right], \tag{3.7}$$

where we have also used x_u for the u-th entry of $X_{\mathcal{Q}}$ to keep the notation simple. We see that the only parameters involved in the conditional distribution of x_u are $\{\Delta_{uv}\}_{v\in\mathcal{Q}}$ and $\{\phi_{iu}\}_{i\in\mathcal{C}}$.

The matrices Θ, Δ and Φ can thus be considered as weighted adjacency matrix respectively for the subgraphs containing only edges between categorical variables, quantitative variables, and edges linking categorical and quantitative variables:

- ϕ_{iu} for $i \in \mathcal{C}$ and $u \in \mathcal{Q}$ is the weight corresponding to the edge potential between the quantitative variable x_u and the categorical variable x_i. We clearly see by (3.6) and (3.7) that the conditional distribution of x_i depends on x_u if and only if $\Phi_{iu} \neq 0$, with the same condition for the conditional distribution of x_i. Hence there is an edge in the graph between the nodes associated to the variables x_i and x_j if and only if $\Phi_{iu} \neq 0$.

- θ_{ij} for $i, j \in \mathcal{C}$ is the weight corresponding to the edge potential between the categorical variables x_i and x_j. We see by (3.6) that the conditional distribution of x_i depends on x_j if and only if $\theta_{ij} \neq 0$. Hence there is an edge in the graph between the nodes associated to the variables x_i and x_j if and only if $\theta_{ij} \neq 0$.

- Δ_{uv} for $u, v \in \mathcal{Q}$ is the weight corresponding to the edge potential between the quantitative variables x_u and x_v. We see by (3.7) that the conditional distribution of x_u depends on x_v if and only if $\delta_{uv} \neq 0$. Hence there is an edge in the graph between the nodes associated to the variables x_u and x_v if and only if $\delta_{uv} \neq 0$.

Example 3.1 The Figure 3.2 illustrates the structure of a mixed graphical models, with four binary variables $x_1, \ldots, x_4$ and three quantitative variables x_5, x_6, x_7, represented by the nodes 1 to 7. In the pairwise undirected model framework, the features are directly readable from the graph structure, and the zeros of the parameters Θ, Δ and Φ are known:

$$\Theta = \begin{bmatrix} ? & ? & 0 & 0 \\ ? & ? & ? & 0 \\ 0 & ? & ? & 0 \\ 0 & 0 & 0 & ? \end{bmatrix}, \qquad \Delta = \begin{bmatrix} ? & ? & ? \\ ? & ? & 0 \\ 0 & ? & ? \end{bmatrix}, \qquad \Phi = \begin{bmatrix} ? & 0 & 0 \\ 0 & 0 & 0 \\ ? & ? & 0 \\ ? & ? & 0 \end{bmatrix},$$

where the ? indicates non-zero elements of the matrices. Note that we did not show the parameter μ, since it is not related to the structure of the graph.

3.1.4 A sampling algorithm

The proposition 3.1 suggests an algorithm to sample from the mixed distribution (3.4). By the Bayes theorem, we have the decomposition

$$p_\Omega(x_\mathcal{C}, x_\mathcal{Q}) = p_\Omega(x_\mathcal{C}) p_\Omega(x_\mathcal{Q}|x_\mathcal{C}).$$

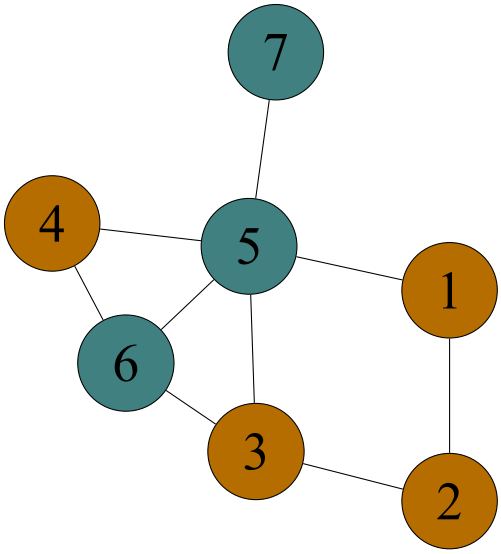


Figure 3.2: Structure of a mixed graphical model. The model has four binary variables x_1, x_2, x_3 and x_4, represented by the brawn nodes with numbers 1 to 4, and three quantitative variables x_5, x_6 and x_7, represented by the grey nodes with number 5, 6 and 7.

The property 3.1.iii states that the marginal distribution of $x_{\mathcal{C}}$ is an Ising model for which the Wolff clustering algorithm can be used (Wolff [1989]). This algorithm is a variant of the classic Swendsen-Wang algorithm (see Barbu and Zhu [2005]) and has good mixing properties. An adapted version of the Wolff algorithm to sample from an Ising model is proposed in Algorithm 1.

Once we defined the Algorithm 1 to sample from the marginal distribution $p_\Omega(x_{\mathcal{C}})$ of the binary variables, we propose the Algorithm 2 to sample a Markov chain from the mixed density (3.4). This algorithm uses the Wolff algorithm to sample a Markov chain $\{X_{\mathcal{C}}^{(j)}\}_j$ from $p_\Omega(x_{\mathcal{C}})$ and exploits the property 3.1.i to sample from the conditional distribution $p_\Omega(x_{\mathcal{Q}}|x_{\mathcal{C}})$. In practice, the algorithm can be initialised randomly, and the first samples of the generated Markov chain have to be discarded to avoid the starting samples over-sampling low-probability regions. In high dimension, one might also consider discarding samples between two accepted samples, otherwise the sampling might over-samples a region of the distribution.

Algorithm 1 Wolff algorithm to sample from the Ising model (3.3)

Input A starting sample $X^{(0)} \in \{0,1\}^{|\mathcal{C}|}$ and an Ising model (3.3) parametrised by a symmetric matrix Θ with dimension $|\mathcal{C}| \times |\mathcal{C}|$, and a Markov chain length M.

for $t = 0, \dots, M-1$ **do**
  Select randomly $i \in \{1, \dots, |\mathcal{C}|\}$ and set $C = \{i\}$,
  Set the set of visited nodes $V = \{i\}$.
  while V is not empty **do**
    Remove an element j from V,
    for $j' \in \mathcal{C}$ such that $j' \notin C$ and $\theta_{jj'} > 0$ **do**
      With probability $1 - \exp(-\theta_{jj'})$, add j' to C and to V.
  Denote by $\bar{X}_i$ the complementary of X_i in $\{0,1\}$,
  Create a new vector $\tilde{X} \in \{0,1\}^{|\mathcal{C}|}$ where $\tilde{X}_j = \bar{X}_i^{(t)}$ for $j \in C$ and $\tilde{X}_j = X_j^{(t)}$ for $j \notin C$.
  Define $X^{(t+1)} = \tilde{X}$ with probability

$$\min\left(1, (\bar{X}_i - X_i) \sum_{j \in C_0} \theta_{jj}\right),$$

  and $X^{(t+1)} = X^{(t)}$ with remaining probability.

Return the sampled sequence $(X^{(t)})_{t=0,\dots,M-1}$.

Algorithm 2 Sampling from the mixed distribution (3.4)

Input a starting sample X^0, a mixed model (3.4) parametrised by $\Omega = (\Theta, \mu, \Delta, \Phi)$, and a Markov chain length M.

Sample a Markov chain $(X_{\mathcal{C}}^{(t)})_{t=0,\dots,M-1}$ from the Ising model with parameters $\Theta + \Phi\Delta^{-1}\Phi^T/2 + \text{Diag}\,(\Phi\Delta^{-1}\mu)$ using the algorithm 1,
For each $t = 0, \dots, M-1$, given $X_{\mathcal{C}}^{(t)}$, sample a Gaussian vector $X_{\mathcal{Q}}^{(t)}$ from the Gaussian distribution with mean $\Delta^{-1}\left(\mu + \Phi^T x_{\mathcal{C}}\right)$ and covariance matrix Δ^{-1}.

Return $(X^{(m)} = (X_{\mathcal{C}}^{(m)}, X_{\mathcal{Q}}^{(m)}), m = 0 \dots M-1)$.

3.2 Model learning

In this section, we will present two algorithms for learning a mixed graphical model. This learning will be done by optimising a penalised likelihood function using a proximal gradient algorithm. We address the calculation of the partition term (3.5) in a first approach by approximating it using MCMC simulations, and in a second approach by optimising a pseudo-likelihood function instead of the classic likelihood: the pseudo-likelihood can actually be expressed in closed-form and do not require a numerical approximation for the likelihood.

3.2.1 The model learning problem

First, we will discuss the general learning problem, which consists in optimising a penalised likelihood function of the parameters Ω given a training set of data.

3.2.1.1 The concavity of the likelihood function

We first explore more in details the likelihood function in the log-linear framework (2.1). In particular, we show that this function is a concave function over a continuous space of parametrisation, what will allow many optimisation techniques to be used.

Given a set of M samples $\mathcal{D} = \{X^{(m)} = (X_{\mathcal{C}}^{(m)}, X_{\mathcal{Q}}^{(m)}), m = 1, \dots, M\}$, the log-likelihood of $\Omega = (\Theta, \mu, \Delta, \Phi)$ is given by

$$\begin{aligned}\ell(\Omega : \mathcal{D}) &= \frac{1}{M} \sum_{m=1}^{M} \log p_\Omega(X^{(m)}) \\ &= \frac{1}{M} \sum_{m=1}^{M} \left[x_{\mathcal{C}}^T \Theta x_{\mathcal{C}} + \mu^T x_{\mathcal{Q}} - \frac{1}{2} x_{\mathcal{Q}}^T \Delta x_{\mathcal{Q}} + x_{\mathcal{C}}^T \Phi x_{\mathcal{Q}} \right] - \log Z_\Omega. \end{aligned} \tag{3.8}$$

Lets take a closer look at the log-partition function $\log Z_\Omega$. We start by introducing a new formulation of the mixed model using a general notation for the features and the parameters:

$$\begin{aligned} p_\Omega(x) &= \frac{1}{Z_\Omega} \exp \left[x_{\mathcal{C}}^T \Theta x_{\mathcal{C}} + \mu^T x_{\mathcal{Q}} - \frac{1}{2} x_{\mathcal{Q}}^T \Delta x_{\mathcal{Q}} + x_{\mathcal{C}}^T \Phi x_{\mathcal{Q}} \right] \\ &= \frac{1}{Z_\Omega} \exp \left[\sum_{k=1}^{K} \omega_k f_k \right], \end{aligned}$$

where ω_k, $k = 1 \dots K$ is a general notation for every parameters of the model, i.e., members of $\{\theta_{ij}\}_{i \geq j}$, $\{\Delta_{uv}\}_{u \geq v}$, $\{\phi_{iu}\}_{iu}$ or $\{\mu_u\}_u$, and f_k, $k = 1 \dots K$ is a general notation for every features of the model, i.e., $\{x_i\}_{i \in \mathcal{C}}$, $\{x_i x_j\}_{i > j \in \mathcal{C}}$, $\{x_u\}_{u \in \mathcal{Q}}$, $\{-\frac{1}{2} x_u x_v\}_{u \geq v \in \mathcal{Q}}$ and $\{x_i x_u\}_{i \in \mathcal{C}, u \in \mathcal{Q}}$. Note that this notation defines a map between the parameters and the features, as each feature f_k

is associated to a unique parameter ω_k, and reciprocally. Using this notation, the first derivative of $\log Z_\Omega$ with regard to any parameter ω_{k_0} is given by

$$\begin{aligned}\frac{\partial}{\partial \omega_{k_0}} \log Z_\Omega &= \frac{1}{Z_\Omega} \sum_{x_\mathcal{C} \in \{0,1\}^{|\mathcal{C}|}} \int_{\mathbb{R}^{|\mathcal{Q}|}} \frac{\partial}{\partial \omega_{k_0}} \exp \sum_k \omega_k f_k(x_\mathcal{C}, x_\mathcal{Q}) dx_\mathcal{Q} \\ &= \frac{1}{Z_\Omega} \sum_{x_\mathcal{C} \in \{0,1\}^{|\mathcal{C}|}} \int_{\mathbb{R}^{|\mathcal{Q}|}} f_{k_0}(x_\mathcal{C}, x_\mathcal{Q}) \exp \sum_k \omega_k f_k(x_\mathcal{C}, x_\mathcal{Q}) dx_\mathcal{Q} \\ &= \mathbb{E}_\Omega[f_{k_0}].\end{aligned}$$

Similarly, we can consider the second derivative:

$$\begin{aligned}\frac{\partial^2}{\partial \omega_i \partial \omega_j} \log Z_\Omega &= \frac{\partial}{\partial \omega_j} \left[\frac{1}{Z_\Omega} \sum_{x_\mathcal{C} \in \{0,1\}^{|\mathcal{C}|}} \int_{\mathbb{R}^{|\mathcal{Q}|}} f_i(x_\mathcal{C}, x_\mathcal{Q}) \exp \sum_k \omega_k f_k(x_\mathcal{C}, x_\mathcal{Q}) dx_\mathcal{Q} \right] \\ &= -\frac{1}{Z_\Omega^2} \left(\frac{\partial}{\partial \omega_j} Z_\Omega \right) \sum_{x_\mathcal{C} \in \{0,1\}^{|\mathcal{C}|}} \int_{\mathbb{R}^{|\mathcal{Q}|}} f_i(x_\mathcal{C}, x_\mathcal{Q}) \exp \sum_k \omega_k f_k(x_\mathcal{C}, x_\mathcal{Q}) dx_\mathcal{Q} \\ &\quad + \frac{1}{Z_\Omega} \sum_{x_\mathcal{C} \in \{0,1\}^{|\mathcal{C}|}} \int_{\mathbb{R}^{|\mathcal{Q}|}} f_i(x_\mathcal{C}, x_\mathcal{Q}) f_j(x_\mathcal{C}, x_\mathcal{Q}) \exp \sum_k \omega_k f_k(x_\mathcal{C}, x_\mathcal{Q}) dx_\mathcal{Q} \\ &= -\frac{1}{Z_\Omega} \left(\frac{\partial}{\partial \omega_j} Z_\Omega \right) \mathbb{E}_\Omega[f_i] + \mathbb{E}_\Omega[f_i f_j] = \mathbb{E}_\Omega[f_i f_j] - \mathbb{E}_\Omega[f_i] \mathbb{E}_\Omega[f_j] \\ &= \mathrm{Cov}_\Omega(f_i, f_j).\end{aligned}$$

We thus see that the Hessian of $\log Z_\Omega$ is the covariance matrix of the features $f_k(x_\mathcal{C}, x_\mathcal{Q})$ viewed as random variables according to the distribution p_Ω. Since a covariance matrix is always positive semi-definite, it follows that the negative log-partition function is a convex function of the parameters Ω. Since the log-likelihood is the sum of a concave term and a linear term in the parameters Ω, we proved that the likelihood function is a convex function.

Proposition 3.2 *The log-likelihood function* (3.8) *is a concave function in the parameters* Ω.

Note that this proposition actually holds for any distribution that is a member of the log-linear framework (2.1), with the same proof.

This result implies that the likelihood has no local optimum. However, it does not imply the uniqueness of the global optimum, and the model may still not be identifiable from data. In particular, several parametrisations might give rise to the same distribution, and such parametrisations are called *redundant parametrisations* (see Friedman and Koller [2009]). A necessary condition for having redundant parameters is the following: in the case of a log-linear represen-

Redundant parametrisation

tation, there are coefficients $\alpha_0, \ldots, \alpha_K$ such that

$$\alpha_0 + \sum_k \alpha_k f_k = 0, \tag{3.9}$$

for almost all features $f = (f_k)_k$ from the feature space, with respect to the common dominating measure χ. This is a necessary and sufficient condition to have $p_\Omega = p_{\Omega'}$ with $\Omega' = \{\Omega_1 + \alpha_1, \ldots, \Omega_k + \alpha_k\}$. Now we observe that the Hessian of the log-likelihood is positive definite if and only if the covariance matrix $(\mathrm{Cov}_\Omega(f_i, f_j))_{i,j}$ is positive definite, which is exactly the negation of the condition (3.9). Hence we conclude that the log-likelihood is strictly concave if and only if the model is identifiable.

In the case of our mixed model framework, the features are $\{x_i\}_{i\in\mathcal{C}}$, $\{x_i x_j\}_{i>j\in\mathcal{C}}$, $\{x_u\}_{u\in\mathcal{Q}}$, $\{-\frac{1}{2}x_u x_v\}_{u\geq v\in\mathcal{Q}}$ and $\{x_i x_u\}_{i\in\mathcal{C},u\in\mathcal{Q}}$. The condition (3.9) can be reformulated as

$$\alpha_0 + \sum_{i,j\in\mathcal{C}} \alpha_{ij} x_i x_j + \sum_{u\in\mathcal{Q}} \alpha_u x_u - \frac{1}{2}\sum_{u,v\in\mathcal{Q}} \alpha_{uv} x_u x_v + \sum_{i\in\mathcal{C},u\in\mathcal{Q}} \alpha_{iu} x_i x_u = 0. \tag{3.10}$$

To prove that the mixed model (3.4) is identifiable from data, one needs to verify that the equation (3.10) holds for almost every vector x only if $\alpha = 0$, with respect to the counting measure for $x_\mathcal{C} = (x_i, i \in \mathcal{C})$ and to the Lebesgue measure for $x_\mathcal{Q} = (x_u, u \in \mathcal{Q})$. In particular, $x_\mathcal{C}$ takes a finite number of values, so the equation (3.10) holds for any specific assignment for $x_\mathcal{C}$. Considering the case $x_i = 0$ for all $i \in \mathcal{C}$ leads to the following equation:

$$\alpha_0 + \sum_{u\in\mathcal{Q}} \alpha_u x_u - \frac{1}{2}\sum_{u,v\in\mathcal{Q}} \alpha_{uv} x_u x_v = 0.$$

This equation corresponds to the identifiable condition (3.9) for a Gaussian model. However, it is well known that a Gaussian model is identifiable from data. Hence it results that $\alpha_0 = 0$, $\alpha_u = 0$ and $\alpha_{uv} = 0$, for all $u, v \in \mathcal{Q}$. The equation (3.10) now boils down to

$$\sum_{i,j\in\mathcal{C}} \alpha_{ij} x_i x_j + \sum_{i\in\mathcal{C},u\in\mathcal{Q}} \alpha_{iu} x_i x_u = 0.$$

By considering $x_{i_0} = 1$ for $i_0 \in \mathcal{C}$ and $x_i = 0$ for all $i \in \mathcal{C}$ with $i \neq i_0$, we have that

$$\alpha_{i_0,u} x_u = 0,$$

which has to hold for almost every x_u with respect to the Lebesgue measure. This implies that $\alpha_{i_0,u} = 0$ for all $u \in \mathcal{Q}$, and a fortiori that $\alpha_{iu} = 0$ for all $i \in \mathcal{C}, u \in \mathcal{Q}$. Hence, equation (3.10)

now boils down to

$$\sum_{i,j\in\mathcal{C}} \alpha_{ij} x_i x_j = 0.$$

Similarly, by fixing $x_{i_0} = 1, x_{j_0} = 1$ and $x_i = 0$ for all $i \neq i_0, j_0$, we get that $\alpha_{i_0,j_0} + \alpha_{j_0,i_0} = 0$. Since $(\alpha_{ij})_{i,j}$ is symmetric, we get that $\alpha_{ij} = 0$ for all $i, j \in \mathcal{C}$. We have thus proved that the mixed model is identifiable from data, and thus that the likelihood is a strictly concave function.

Proposition 3.3 *The log-likelihood* (3.8) *is a strictly concave function in the parameters* Ω.

3.2.1.2 Model learning with regularisations

We have shown that the likelihood function is a concave function over a continuous space of parameters and that the global optima of the likelihood is unique. However, as discussed in section 2.4.4.3, the maximum likelihood estimator is often prone to overfitting to the data and results in a fully connected network.

The usage of parameters priors $p(\Omega)$ can reduce the effect of overfitting, as it was the case for Bayesian networks. Most commonly used priors include Gaussian priors and Laplacian priors, leading respectively to the well known ℓ_2 and ℓ_1 regularisations.

ℓ_2-regularisation **The ℓ_2-regularisation** (e.g., Nigam et al. [1999]) involves the use of a zero-mean Gaussian prior on the log-linear parameters Ω

$$p(\Omega|\sigma^2) = \prod_{i=1}^{k} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\omega_i^2}{2\sigma^2}\right),$$

for a fixed variance σ^2. Converting to log-space, using this prior gives rise to the term

$$-\frac{1}{2\sigma^2}\sum_{i=1}^{k} \omega_i^2 = -\frac{1}{2\sigma^2}||\Omega||_2^2,$$

where $||\cdot||_2$ is the L_2-norm. Note that this term is concave.

ℓ_1-regularisation **The ℓ_1-regularisation,** also called Lasso regularisation (see Tibshirani [1996]), involves the use of a zero-mean Laplacian prior

$$p(\Omega|\beta) = \prod_{i=1}^{k} \frac{1}{2\beta} \exp\left(-\frac{|\omega_i|}{\beta}\right).$$

In the log-space, this prior gives rise to the term

$$-\frac{1}{\beta}\sum_{i=1}^{k}|\omega_i| = -\frac{1}{\beta}||\Omega||_1,$$

where $||\cdot||_1$ is the L_1-norm. Note that this term is also concave.

ℓ_1/ℓ_2-regularisation

The ℓ_1/ℓ_2-regularisation, also called Group Lasso regularisation (Yuan and Lin [2006]) is a special case of the Lasso regularisation where knowledge about the structure of the variables is known a priori. Suppose our variables are divided into J groups of variables, whose indices are denoted by $K_1, \ldots, K_J$, then the Group Lasso regularisation involves the term

$$-\frac{1}{\beta}\sum_{j=1}^{J}||\omega_{K_j}||_2,$$

where ω_{K_j} denotes the subset $\{\omega_i, i \in K_j\}$. Note that this term is also concave.

All regularisations penalise parameters with a high magnitude (positive or negative), however their impacts are very different. Many authors have discussed about the effects of their use (e.g., Ng [2004]). In practice, the main difference comes from the fact that the models learned with a ℓ_1-regularisation tend to be much sparser than models learned with ℓ_2- or ℓ_1/ℓ_2-regularisation, i.e., models where a lot of parameters are null. From a structural perspective, this results in a graphical model with fewer edges and sparser potentials.

Note also that both regularisations are concave, and because the log-likelihood is also concave, the posterior can thus be optimised using gradient-based methods. For this purpose, the use of regularisation has been widely studied over the last years, especially for ℓ_1 and ℓ_1/ℓ_2 regularisations, see Schmidt [2010] for recent reviews and Varoquaux et al. [2010] for the use of ℓ_1/ℓ_2 in graph structure estimation.

Remember that an Ising model can only model binary variables, whereas our training data includes categorical non-binary variables and that this non-binary variables will be transformed into a binary vector using the 1-to-K encoding scheme (see section 3.1.1). In the following, we will suppose that this transformation has already been done – i.e., all the categorical variables are binary – and we denote by G the set of variables indices arising from this transformation, i.e., the indices from the binary variables originated from the same categorical non-binary variable.

The general learning problem we want to solve is finding the estimator

$$\hat{\Omega} = \underset{\Omega}{\text{Argmin}}\left(-\ell(\Omega : \mathcal{D}) + g(\Omega)\right), \tag{3.11}$$

where $\mathcal{D} = \{X^{(1)}, \ldots, X^{(M)}\}$ is a training set and g is a penalisation function over the parame-

ters Ω.

Since the Lasso penalty tends to provide sparser structures, we will use a particular penalty that involves group Lasso and Lasso regularisation for the parameters corresponding to edges parameters, i.e., Θ, Δ and Φ (Bach et al. [2012]). Remember from section 2.4.5.1 that in the framework of undirected models, the absence of edge between two nodes x_i and x_j states that they are conditionally independent given all the others variables.

In addition, we add a compact constraint on Δ, so that Δ remains inside a compact set included in the cone of positive definite matrices. That hypothesis is used to ensure that p_Ω is a valid density (see definition 3.1), but also for numerical reasons. Having a positive-definite precision matrix inside a compact set indeed ensures that our learning criterion is gradient-Lipschitz, which is a required hypothesis for the learning algorithms we will use in the next sections.

The compact constraint is defined as follow: for any $0 < \rho < 1$, denote by $\mathcal{K}_\rho$ the compact subset of positive definite symmetric matrices defined by

$$\mathcal{K}_\rho = \{\Delta_0^{1/2}(I+\epsilon)\Delta_0^{1/2} \ : \ \epsilon \text{ is symmetric with } -\rho < \lambda_{\min}(\epsilon) < \rho\},$$

where I is the identity matrix, $\lambda_{\min}$ denotes the minimal eigenvalue and $\lambda_{\max}$ denotes the maximal one and Δ_0 is the empirical precision defined by

$$\Delta_0 = \left[\frac{1}{M}\sum_{m=1}^{M}(X^{(m)} - \bar{X})(X^{(m)} - \bar{X})^T\right]^{-1},$$

where $\bar{X}$ denotes the empirical mean of the set $\{X^{(m)}, m = 1 \dots M\}$ of M samples. Observe that $\mathcal{K}_\rho$ is a closed convex set of symmetric matrices containing a ball centred at Δ_0. Here ρ is arbitrary chosen to ensure the convergence of the numerical optimization. In practice, one needs to check that the obtained optimizer is in the interior of the compact set.

The penalisation we use is

$$g(\Omega) = \lambda_\theta \sum_{k \neq k' \in K} \|\theta_{kk'}\|_2 + \mathbb{I}_{\{\mathcal{K}_\rho\}}(\Delta) + \lambda_\Delta \sum_{u<v \in \mathcal{Q}} |\Delta_{uv}| + \lambda_\Phi \sum_{k \in K, u \in \mathcal{Q}} \|\Phi_{ku}\|_2, \tag{3.12}$$

where $\mathbb{I}_{\{\mathcal{K}_\rho\}}$ is the characteristic function of the closed convex set $\mathcal{K}_\rho$, i.e.,

$$\mathbb{I}_{\{\mathcal{K}_\rho\}}(\Delta) = \begin{cases} 0 & \text{if } \Delta \in \mathcal{K}_\rho, \\ +\infty & \text{otherwise,} \end{cases}$$

where $\theta_{kk'} = (\theta_{ii'})_{i \in k, i' \in k'}$ and $\phi_{ku} = (\phi_{iu})_{i \in k}$ where, for all $i \in \mathcal{C}$, k_i is the set of indices of

binary variables created after applying 1-of-K scheme over non binary categorical variable x_i. Notice that we do not penalise the diagonal terms of Θ and Δ.

3.2.2 Related works mixed model learning

The problem of optimising on concave regularised likelihoods has attracted some attention over the last years, see Schmidt [2010] and Bach et al. [2012] for quite exhaustive surveys on that topic.

In the case where the variables are supposed to be drawn from a multivariate zero-mean Gaussian distribution $\mathcal{N}(0, \Sigma)$ where the true covariance matrix Σ is unknown, the learning problem is called covariance selection and is a widely studied topic. Lauritzen [1996] uses a forward-backward search algorithm to determine the zeros in the precision matrix, but this approach do not scale with the dimension and appears intractable even for a moderate number of variables. Li and Gui [2006] uses a gradient descent algorithm over an objective function that is the negative log-likelihood that takes into account the sparsity of the precision function. d'Aspremont et al. [2008] solves a maximum likelihood problem penalized by the number of non-zeros in the inverse covariance matrix, in order to find a sparse representation of the data and to discover conditional independences between the variables.

The learning of Ising or Potts models, used for categorical and binary variables, has been applied in a large variety of fields, see e.g., Wainwright and Jordan [2008] for applications in several areas associated to categorical variables. Loh et al. [2013] investigate the problem of the relationship between conditional independences for categorical variables and the structure of the inverse covariance matrix. They show that the structure is reflected in the inverse covariance matrix of an augmented sets of variables that include higher-order interactions between variables. Many authors consider the optimisation of a penalised likelihood. However, contrary to the Gaussian models, the partition function of an Ising or Potts model is intractable and its evaluation is NP-hard (see, e.g., Friedman and Koller [2009]). Wainwright et al. [2006] deals with this intractability by fitting ℓ_1-regularised logistic regressions on the conditional distributions. Schmidt et al. [2008] consider a ℓ_1-penalised symmetric pseudo-likelihood estimation, where the conditional probability distributions of each variable given the others is seen as the likelihood of a logistic regression model.

The general problem of optimising a penalised likelihood function has motivated several works. Among them, the subgradient descent approach is widely applicable (see, e.g., Bertsekas [1999]). It is an iterative scheme for learning with a low running time complexity, but also a slow convergence rate. Namely, each iteration consists of the computation $\Omega^{(t+1)} = \Omega^{(t)} - \frac{\alpha}{t^\beta}(s + \lambda s')$, where $s \in \partial\ell(\Omega^{(t)})$ and $s' \in \partial g(\Omega^{(t)})$. Coordinate descent methods (see e.g., Fu [1998]) is based on the optimisation of one coordinate at a time. It is also an iterative scheme: at each iteration t,

the algorithm solves the optimisation problem $\omega_i^{(t+1)} = \underset{y}{\text{argmin}} \ell(\omega_1^{(t)}, \ldots, \omega_{i-1}^{(t)}, y, \omega_{i+1}^{(t)}, \ldots, \omega_k^{(t)})$. This method is effective if ℓ is close to be separable, but the performance deteriorates as the dependences between the variables increase.

Another way to recover the structure of a graphical model is to determine the set of neighbours of each node in the graph by regressing each variables against all the other remaining variables. This method, sometimes referred to as node-wise regression problems, have been widely studied. Tibshirani [1996] uses the Lasso to discover a short list neighbours of each node in the graph. Meinshausen and Bühlmann [2006] studied this approach in the case of Gaussian model learning and showed that the used estimator is consistent, even in high dimension. Namely, they are fitting a Lasso model to each variables and estimate each entry Δ_{ij} of the precision matrix to be non-zero if either the estimated coefficient of the variable x_i on x_j or the estimated coefficient of x_j on x_i is non-zero, i.e., they use a "and" rule. Ravikumar et al. [2010] separately uses the same technique for estimating Ising models structures. Friedman et al. [2008] proposes a similar approach, the well known graphical Lasso, that uses a coordinate gradient descent to maximise the objective function $\log\det\Delta - \text{Tr}\,(S\Delta) - \lambda\|\Delta\|_1$, where S is the empirical covariance matrix and Δ is the precision matrix over which the objective function is maximised.

Mixed graphical models have attracted a few studies over the last years. Some approaches are transforming the data, for instance Bach and Jordan [2002] proposes a solution where the variables used for the learning are Gaussian versions of the initial data, whether they are categorical or quantitative, using Mercer kernels.

Lauritzen [1996] proposes a graphical model framework with the property that, conditioned to the categorical variables $x_\mathcal{C}$, the quantitative variables $x_\mathcal{Q}$ have a Gaussian density with a mean and a covariance matrix depending on $x_\mathcal{C}$. This model is a more general framework than the model (3.4), but has a number of parameters that scales exponentially with the number of categorical variables and is not suitable for high dimension problems: to each value assignments of the categorical variables is associated a parametrisation of a Gaussian density.

In Laby et al. [2015], we introduce the mixed graphical model (3.4), which is a model that mixes an Ising and a Gaussian model. This model has been separately studied by Lee and Hastie [2015]. Haslbeck and Waldorp [2015b] proposes a generalisation of the generalised covariance approach of Loh et al. [2013] to estimate the structure of the mixed same graphical model. This model is a simplified version of the mixed model proposed by Cheng et al. [2013], which allows higher-order interactions, where the covariances of the quantitative variables are functions of the categorical variables.

Tur and Castelo [2012] addresses the learning problem by proposing a method based on limited-order correlations that are used to determine conditional dependences between variables, where the size of the conditioning set of variables is bounded. This method has been designed for the case where the number of samples M is much lower than the number of variables n. In order

for the likelihood estimates to exist, the authors have made the assumption that the variables are marginally independent.

Node-wise regression algorithms have been also studied, where a linear model is used to estimate $p(x_i|x_{-i})$ for all variables x_i. This method has been separately applied by Meinshausen and Bühlmann [2006] for Gaussian model learning and by Ravikumar et al. [2010] for Ising models, and Lee and Hastie [2015] proposes a version for learning mixed models. Yang et al. [2012] and Yang et al. [2013] are also using node-wise regressions to learn generalized mixed model, where the conditional distributions of each variables are members of the exponential family.

To address the calculation of the partition function (3.5), Laby et al. [2015] and Lee and Hastie [2015] are using pseudo-likelihood (Besag [1975]) (with different formulation of the pseudo-likelihood) instead of the likelihood. This method is computationally efficient since the pseudo-likelihood can be expressed in closed-form, but is sub-optimal.

Proximal methods are a class of methods that particularly suit learning problems when the objective function is the sum of a smooth term and a non-smooth term. The use of the ℓ_1-regularisation makes the objective function non smooth. Lee and Hastie [2015] uses proximal gradient and proximal Newton algorithm to minimise a pseudo-likelihood. In the following section, we will describe a stochastic version of the proximal gradient.

3.2.3 The proximal gradient algorithm

In this section, we present the proximal gradient class of algorithms for solving convex optimisation problems. While Newton's algorithms are standard methods for solving smooth unconstrained minimisation problems, proximal algorithms (Bach et al. [2012]; Parikh and Boyd [2013]) are tools for solving non-smooth, constrained or large-scale version of these problems. They are specially designed for solving problems that can be formulated as a sum of a smooth differentiable function with Lipschitz-continuous gradient and a non-differentiable function, as it is the case in our problem form (3.11).

Proximal gradient algorithms are based on an iterative scheme, where at each iteration the evaluation of a proximal operator of the objective function is done. The computation of the proximal operator involves solving a convex optimisation problem. This sub-problem is often easy to solve because a solution can be expressed in closed-form.

Proximal operator

The *proximal operator* $\text{Prox}_g : \mathbb{R}^n \to \mathbb{R}^n$ of a closed proper convex function $g : \mathbb{R}^n \to \mathbb{R}$ is defined by

$$\text{Prox}_g(v) = \underset{x}{\text{argmin}} \left(g(x) + \frac{1}{2}||x - v||_2^2 \right),$$

where $||\cdot||_2$ is the usual L_2-norm. We often encounter the case where the used function is a scaled function βf, where $\beta > 0$. In that case, the proximal operator can be expressed as

$$\text{Prox}_{\beta g}(v) = \underset{x}{\text{argmin}} \left(g(x) + \frac{1}{2\beta}||x - v||_2^2 \right). \tag{3.13}$$

The proximal operator defines a map from each vector $v \in \mathbb{R}^n$ to the unique solution $\text{Prox}_{\beta g}(v)$ of the minimisation problem. Note that this solution is actually unique since the sum of the convex function g and the strictly convex squared Euclidean norm defines a strictly convex function. The parameter λ controls the compromise between minimising g and minimising $||x-v||_2^2$: larger values tends to move the proximal point toward the minimum of g, whereas lower values tends to move $\text{Prox}_{\beta g}(v)$ closer from v. $\text{Prox}_{\beta g}(v)$ is thus called *proximal point* of v with respect to βg. The Figure 3.3 illustrates the concept of proximal point in the case where $n = 1$ and g is the absolute value function.

Proximal point

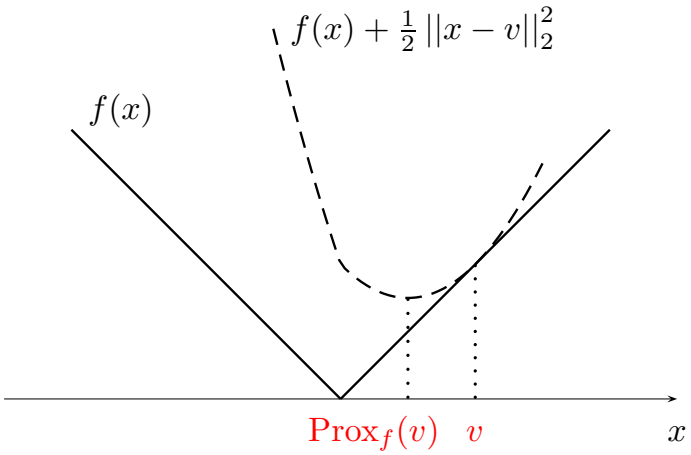


Figure 3.3: Evaluation of the proximal point $\text{Prox}_g(v)$ of the point v with respect to the function $g : x \mapsto |x|$. The proximal point $\text{Prox}_g(v)$ is minimising the function $x \mapsto |x| + \frac{1}{2}||x - v||_2^2$.

Having a closed-form of the proximal operator allows a fast computation of the proximal point and leads to fast convergence rate of proximal methods. There are many cases where such a closed-form can be calculated (see Parikh and Boyd [2013]). The ones we encounter in this thesis are listed below.

In the case where g is the characteristic function $\mathbb{I}_{\{\mathcal{C}\}}(\Omega)$ of a closed convex set $\mathcal{C}$, i.e.,

$$\mathbb{I}_{\{\mathcal{C}\}}(\Omega) = \begin{cases} 0 & \text{if } \Omega \in \mathcal{C}, \\ +\infty & \text{otherwise,} \end{cases}$$

the proximal operator reduces to the orthogonal projection on $\mathcal{C}$, i.e., the function $\Pi_{\mathcal{C}}(v)$ defined by

$$\Pi_{\mathcal{C}}(v) = \underset{\Omega \in \mathcal{C}}{\operatorname{argmin}} ||\Omega - v||^2. \tag{3.14}$$

In the case where f is the ℓ_1-regularisation $\beta\|\Omega\|_1 = \beta \sum_i |\omega_i|$**,** the proximal operator reduces to the component-wise soft-thresholding operator s_β defined, for each component ω_i of Ω, by

$$s_\beta(\omega_i) = \begin{cases} \omega_i - \beta & \text{if } \omega_i \geq \beta, \\ \omega_i + \beta & \text{if } \omega_i \leq \beta, \\ 0 & \text{otherwise.} \end{cases} \tag{3.15}$$

In the case where f is the ℓ_1/ℓ_2-regularisation $\beta \sum_j \|\omega_{K_j}\|_2$**,** the proximal operator reduces to the component-wise soft-thresholding $\tilde{s}_{\beta,K}$ defined, for each subset of parameters $\omega_{K_j} = \{\omega_i, i \in K_j\}$, by

$$\tilde{s}_{\beta,K}(\omega_{K_j}) = \begin{cases} \omega_{K_j} - \beta \frac{\omega_{K_j}}{\left|\left|\omega_{K_j}\right|\right|_2} & \text{if } \left|\left|\omega_{K_j}\right|\right|_2 > \beta, \\ 0 & \text{otherwise.} \end{cases} \tag{3.16}$$

Note that in (3.12), the precision matrix Δ is penalised by the sum

$$\mathbb{1}_{\{\mathcal{K}_\rho\}}(\Delta) + \lambda_\Delta \sum_{u<v \in \mathcal{Q}} |\Delta_{uv}|,$$

Generalised forward-backward splitting

for which there is no simple closed-form. We will address this using the *generalised forward-backward splitting* algorithm Raguet et al. [2013] in section 3.2.4.

The proximal gradient algorithm to minimise (3.11) – i.e., a sum of the opposite of a likelihood function and a penalisation term – is given in algorithm 3. This algorithm involves, at each iteration, a gradient step computation $\omega^{(t)} + \gamma_t \nabla \ell(\omega^{(t)})$ and the calculation of a proximal point with respect to the scaled penalisation $\gamma_t g(\Omega)$, where g is the penalisation defined by (3.12).

The proximal gradient algorithm comes with some properties concerning the consistency and the uniqueness of the solution (see Bauschke and Combettes [2011] and Atchade et al. [2015]).

Proposition 3.4 *For a proper convex function g and a concave gradient-Lipschitz likelihood ℓ with Lipschitz constant L for $\nabla\ell$, for a fixed gradient step size $\gamma \in [0, 1/L[$,*

Algorithm 3 Deterministic proximal gradient algorithm to minimise $-\ell(\Omega : \mathcal{D}) + g(\Omega)$

Input Step sizes γ_t, a starting point $\Omega^{(0)}$,

At each i :teration t, given the current solution $\Omega^{(t)} = (\omega_1^{(t)}, \ldots, \omega_k^{(t)})$,
    Compute the gradient step $\tilde{\omega}^{(t)} = \omega^{(t)} + \gamma_t \nabla \ell(\omega^{(t)})$,
    Compute $\omega^{(t)} = \text{Prox}_{\gamma_t g}(\tilde{\omega}^{(t)})$.
Return the last estimation $\Omega^{(t)}$

1. *For any starting point Ω satisfying the hypothesis listed in definition 3.1, the series $\{\Omega^{(t)}\}_t$ generated by the proximal gradient Algorithm 3 converges to a solution of the minimisation problem (3.11).*

2. *Any minimiser Ω^* of the function $\Omega \mapsto -\ell(\Omega) + g(\Omega)$ satisfies*

$$\Omega^* = \text{Prox}_{\gamma g}\left(\Omega^* + \gamma \nabla \ell(\Omega^*)\right).$$

There are many ways to interpret the proximal gradient Algorithm 3, and Beck and Teboulle [2009] and Parikh and Boyd [2013] propose several interpretations. Here we explain the majorisation - minimisation approach proposed by Beck and Teboulle [2009], a large class of algorithms that includes the proximal gradient algorithm, Newton's methods, etc. This approach is based on the minimisation of a dominating function through the iteration of the scheme

$$\Omega^{(t+1)} = \underset{\Omega}{\text{argmin}}\ \varphi(\Omega, \Omega^{(t)}), \tag{3.17}$$

where $\varphi(\cdot, \Omega^{(t)})$ is a convex function dominating a function f where $\varphi(\cdot, \Omega^{(t)})$ is tight to f at $\Omega^{(t)}$, that is, $\varphi(\Omega, \Omega^{(t)}) \geq f(\Omega)$ and $\varphi(\Omega^{(t)}, \Omega^{(t)}) = f(\Omega^{(t)})$. In our case of optimising a penalised likelihood, the objective function f is $-\ell + g$, and we consider the dominating function φ defined by

$$\varphi(\vartheta, \Omega) = -\ell(\Omega) - \langle \nabla \ell(\Omega), \vartheta - \Omega \rangle + \frac{1}{2\gamma_t} ||\vartheta - \Omega||_2^2 + g(\vartheta).$$

We can easily show that for any fixed Ω and any ϑ, $\varphi(\vartheta, \Omega) \geq -\ell(\vartheta) + g(\vartheta)$ when $\gamma_t \in (0, \frac{1}{L}[$, where L is the Lipschitz constant of $-\nabla \ell$. Remember from definition 3.1 that the precision matrix Δ was constrained to be positive semi-definite, and hence the likelihood is gradient-Lipschitz. We can also easily show that for any Ω, $\varphi(\Omega, \Omega) = -\ell(\Omega) + g(\Omega)$, and that $\varphi(\cdot, \Omega)$ is convex. In addition, we have that

$$\text{Prox}_{\gamma_t g}\left(\Omega^{(t)} + \gamma_t \nabla \ell(\Omega^{(t)})\right) = \underset{\vartheta}{\text{argmin}}\ \varphi\left(\vartheta, \Omega^{(t)}\right),$$

hence the algorithm (3.17) can be seen as a majorisation-minimisation algorithm.

The algorithm can actually converge with step sizes smaller than $2/L$ (Parikh and Boyd [2013]), although for step sizes greater than $1/L$ the Majorisation-Minimisation approach can no longer be motivated. In practice, when the Lipschitz constant L is not known, the step sizes γ_t can be found by line search, i.e., their values are chosen at each iteration of the algorithm. A simple line search algorithm is proposed by Beck and Teboulle [2009] and is given in algorithm 4.

Algorithm 4 Line search to choose the step size γ_t

Input $\Omega^{(t-1)}$, γ_{t-1}, and a parameter $\beta \in]0, 1[$,

Define $\gamma = \gamma_{t-1}$.
do
  Let $\vartheta = \text{Prox}_g(\Omega^{(t-1)} + \gamma \nabla \ell(\Omega^{(t-1)}))$,
  Break If $-\ell(\vartheta) \leq \varphi(\Omega, \Omega^{(t-1)})$.
  Update $\gamma := \beta\gamma$.
while
Return $\gamma_t = \gamma$, $\Omega^{(t)} = \vartheta$.

In the next two sections, we will show two algorithms we developed to optimise the penalized likelihood to learn a mixed model from data, by addressing the calculation of the partition function Z_Ω defined by (3.5) through a stochastic approximation step.

3.2.4 Learning a mixed model with stochastic proximal gradient

In this section, we present a stochastic version of the proximal gradient algorithm, where the intractability of the computation of the partition function Z_Ω is addressed by estimating it using MCMC simulations. To simplify the notation, we introduce the sufficient statistic F to rewrite the mixed density $p_\Omega(X)$:

$$p_\Omega(X) = \frac{1}{Z_\Omega} \exp(\langle F, \Omega \rangle),$$

where Z_Ω is the normalisation constant defined in (3.5), where $F = (F_1, F_2, F_3, F_4)$ is a sufficient statistic for X with:

- F_1 is the matrix indexed over $\mathcal{C} \times \mathcal{C}$ defined by $F_1 = X_\mathcal{C} X_\mathcal{C}^T$,
- F_2 is the vector indexed over $\mathcal{Q}$ defined by $F_2 = X_\mathcal{Q}$,
- F_3 is the matrix indexed over $\mathcal{Q} \times \mathcal{Q}$ defined by $F_3 = -\frac{1}{2} X_\mathcal{Q} X_\mathcal{Q}^T$,
- F_4 is the matrix indexed over $\mathcal{C} \times \mathcal{Q}$ defined by $F_4 = X_\mathcal{C} X_\mathcal{Q}^T$,

and where $\langle \cdot, \cdot \rangle$ is the dot product defined by

$$\langle F, \Omega \rangle = \mathrm{Tr}\left(\Theta F_1^T\right) + \mu^T F_2 + \mathrm{Tr}\left(\Delta F_3^T\right) + \mathrm{Tr}\left(\Phi F_4^T\right). \tag{3.18}$$

We denote by $\mathcal{D} = \{X^{(j)}\}_{j=1\ldots M}$ the training dataset of M samples and $F^{(j)}$ their sufficient statistics. With this notation, the log-likelihood function of the parameters Ω given $\mathcal{D}$ becomes

$$\ell(\Omega : \mathcal{D}) = \frac{1}{M} \sum_{j=1}^{M} \langle \Omega, F^{(j)} \rangle - \log Z_\Omega. \tag{3.19}$$

In this section, we address the calculation of the partition function Z_Ω. In particular, the likelihood $\ell(\Omega : \mathcal{D})$ and its derivative $\nabla \ell(\Omega : \mathcal{D})$ are intractable by direct calculation. However, the Fisher identity yields $\mathbb{E}_\Omega[\nabla \ell(\Omega)] = 0$, where $\mathbb{E}_\Omega$ denotes the expectation with respect to the distribution p_Ω. By (3.19), we have that

$$\nabla \log Z_\Omega = \mathbb{E}_\Omega[F] = \mathbb{E}_\Omega[\mathbb{E}_\Omega[F|X_\mathcal{C}]]. \tag{3.20}$$

This result suggests an algorithm to estimate $\nabla \log Z_\Omega$. Since it is not possible to sample directly from p_Ω, we turn to MCMC. Remember from proposition 3.1.i that, given the categorical variables $x_\mathcal{C}$, the quantitative variables $x_\mathcal{Q}$ have a Gaussian density with mean $\Delta^{-1}\left(\mu + \Phi^T x_\mathcal{C}\right)$ and covariance matrix Δ^{-1}. We can then compute the conditional expectations $\mathbb{E}_\Omega[F|X_\mathcal{C}]$ of the sufficient statistic F given an observation of the categorical variables $X_\mathcal{C}$:

$$\begin{aligned}
&\mathbb{E}_\Omega[F_1 \mid X_\mathcal{C}] = F_1, \\
&\mathbb{E}_\Omega[F_2 \mid X_\mathcal{C}] = \mathbb{E}(X_\mathcal{Q} \mid X_\mathcal{C}) = \Delta^{-1}(\mu + \Phi^T X_\mathcal{C}), \\
&\mathbb{E}_\Omega[F_3 \mid X_\mathcal{C}] = -\frac{1}{2}\Delta^{-1} - \frac{1}{2}\mathbb{E}_\Omega(F_2 \mid X_\mathcal{C})\mathbb{E}_\Omega(F_2 \mid X_\mathcal{C})^T, \\
&\mathbb{E}_\Omega[F_4 \mid X_\mathcal{C}] = X_\mathcal{C}\mathbb{E}_\Omega(F_2 \mid X_\mathcal{C})^T.
\end{aligned} \tag{3.21}$$

This formulas lead to the estimation of $\nabla \ell(Z_\Omega)$: if $\{\xi^{(m)}\}_{m=1\ldots\eta}$ are η instances of p_Ω, then

$$\nabla \log Z_\Omega \approx \frac{1}{\eta} \sum_{m=1}^{\eta} \mathbb{E}_\Omega[F|\xi_\mathcal{C}^{(m)}]. \tag{3.22}$$

The proposition 3.1.iii stating that the marginal distribution of $X_\mathcal{C}$ is an Ising model with parameters $\Theta + \Phi\Delta^{-1}\Phi^T/2 + \mathrm{Diag}\left(\Phi\Delta^{-1}\mu\right)$ yield the algorithm 5 to estimate $\nabla \log Z_\Omega$ via MCMC simulations.

The proximal gradient algorithm need to be slightly adapted to include the estimation of the partition function. Atchade et al. [2015] propose a stochastic version of the proximal gradient

Algorithm 5 Estimation of $\nabla \log Z_\Omega$

Input a model parametrisation Ω and a MCMC length m,

1 Simulate η samples $\{\xi^{(m)}\}_{m=1\dots\eta}$ from the marginal distribution $p_\Omega(x_\mathcal{C})$ of the categorical variables using, e.g., the Wolff algorithm 1,
2 Compute the conditional expectation $\mathbb{E}_\Omega[F|\xi^{(m)}] = (\mathbb{E}_\Omega[F_i|\xi^{(m)}],\ i = 1\dots4)$ for each sample $\xi^{(m)},\ j = 1\dots\eta$, using the system of equations (3.21),
3 **Return** the estimation of $\nabla \log Z_\Omega$ given by

$$\nabla \log Z_\Omega \approx \frac{1}{\eta}\sum_{m=1}^{\eta} \mathbb{E}_\Omega[F|\xi_\mathcal{C}^{(m)}].$$

Algorithm 3 that takes this estimation into account. This algorithm uses an estimation H_η of the likelihood gradient $\nabla\ell$ using MCMC simulations, where η is the length of the simulated Markov chain. The decomposition (3.19) rise to the gradient of the log-likelihood

$$\nabla\ell(\Omega : \mathcal{D}) = \frac{1}{M}\sum_{m=1}^{M} F^{(m)} - \nabla \log Z_\Omega,$$

and the estimation (3.22) calculated with the Algorithm 5 leads to an estimation of $\nabla\ell(\Omega : \mathcal{D})$, denoted H_η:

$$H_\eta(\Omega) = \frac{1}{M}\sum_{m=1}^{M} F^{(m)} - \frac{1}{\eta}\sum_{m=1}^{\eta} \mathbb{E}_\Omega[F|\xi_\mathcal{C}^{(m)}], \tag{3.23}$$

where $\{F^{(m)}\}_{m=1\dots M}$ are the sufficient statistics of the training data $\mathcal{D}$ and $\{\xi^{(m)}\}_{m=1\dots\eta}$ are instances sampled from p_Ω. From (3.23) rises the stochastic proximal gradient Algorithm 6 to estimate the parameters of a mixed graphical models from data.

Algorithm 6 Stochastic proximal gradient algorithm for mixed model learning

Input a training dataset $\mathcal{D} = \{X^{(m)}, m = 1\dots M\}$, a series of gradient step sizes $\{\gamma_t\}$, a series of MCMC length $\{\eta_t\}$ and a starting point $\Omega^{(0)}$,

1 Compute the left term of (3.23) given by $\frac{1}{M}\sum_{m=1}^{M} F^{(m)}$.
2 **At each** iteration t :
3 Compute an estimation of $\nabla \log Z_{\Omega^{(t)}}$ using the algorithm 5,
4 Infer the estimation $H_\eta(\Omega)$ of $\nabla\ell(\Omega^{(t)})$ using equation (3.23),
5 Compute $\Omega^{(t+1)} = \text{Prox}_{\gamma_t g}(\Omega^{(t)} + \gamma_t H_\eta)$.

Atchade et al. [2015] provide theoretical results to ensure the convergence of the series $\{\Omega^{(t)}\}_t$ generated by the stochastic proximal gradient Algorithm 6. With the same hypothesis

as for proposition 3.4 and some additional assumptions that are verified in our case, the Algorithm 6 is guaranteed to converges almost surely when $t \to \infty$ toward a fixed-point of the algorithm, i.e., an element of $\{\Omega : \Omega = \text{Prox}_{\gamma g}(\Omega + \gamma \nabla \ell(\Omega))\}$.

Note that in our case where we use the penalisation (3.12) defined by

$$g(\Omega) = \lambda_\theta \sum_{k \neq k' \in K} \|\theta_{kk'}\|_2 + \mathbb{I}_{\mathcal{K}_\rho}(\Delta) + \lambda_\Delta \sum_{u<v \in \mathcal{Q}} |\Delta_{uv}| + \lambda_\Phi \sum_{k \in K, u \in \mathcal{Q}} \|\Phi_{ku}\|_2, \tag{3.24}$$

which is the sum of ℓ_1- and ℓ_1/ℓ_2-regularisations on the edges of the network and a compact constraint on Δ to ensure it remains inside the cone of the positive definite matrices. Remember that there is no closed-formed formulation of the proximal operator, when the used penalty is (3.24). With only ℓ_1 and ℓ_1/ℓ_2 regularisations, the proximal operator of the Lasso and group Lasso part of g can be reformulated as a component-wise soft-threshold $\sigma_{\lambda,\gamma,K}(\Omega)$ defined by

$$\sigma_{\lambda,\gamma,K}(\Omega) = (\tilde{s}_{\lambda_\Theta \gamma,K}(\Theta), s_{\lambda_\Delta \gamma}(\Delta), \tilde{s}_{\lambda_\Phi \gamma,K}(\Phi)),$$

where $s_{\lambda\gamma}$ and $\tilde{s}_{\lambda\gamma}$ have been defined respectively in (3.15) and (3.16), and K is the set of the variables tight together with the 1-to-K encoding.

With only the compact constraint $\mathbb{1}_{\{\mathcal{K}_\rho\}}$, the proximal operator is the orthogonal projection on $\mathcal{K}_\rho$, which is the map $\Pi_{\mathcal{K}_\rho}$ defined in (3.14).

Algorithm 7 Generalised forward-backward splitting algorithm

Input a training set $\mathcal{D} = \{X^{(m)}, m = 1 \dots M\}$, gradient step sizes $\{\gamma_t\}$ and a starting point $\Omega^{(0)}$.

1 Define $\Omega_1 = \Omega^{(0)}$ and $\Omega_2 = \Omega^{(0)}$.
2 **At each** iteration t :
3 Compute $\Omega_1 = \Omega_1 + \text{Prox}_{g_1}\left(2\Omega^{(t)} - \Omega_1 + \gamma_t \nabla \ell(\Omega^{(t)})\right)$, where g_1 is defined by

$$g_1(\Omega) = \lambda_\theta \sum_{k \neq k' \in K} \|\theta_{kk'}\|_2 + \lambda_\Delta \sum_{u<v \in \mathcal{Q}} |\Delta_{uv}| + \lambda_\Phi \sum_{k \in K, u \in \mathcal{Q}} \|\Phi_{ku}\|_2.$$

4 Compute $\Omega_2 = \Omega_2 + \text{Prox}_{g_2}\left(2\Omega^{(t)} - \Omega_2 + \gamma_t \nabla \ell(\Omega^{(t)})\right)$, where g_2 is defined by

$$g_2(\Omega) = \mathbb{I}_{\mathcal{K}_\rho}(\Delta).$$

5 Compute $\Omega^{(t+1)} = \frac{1}{2}(\Omega_1 + \Omega_2)$.
6 **Return** the last computed $\Omega^{(t+1)}$.

Generalised forward-backward splitting

Since our penalization is a sum of two standard penalties, we use the *generalized forward-backward splitting algorithm* Raguet et al. [2013]. This algorithm is designed to minimise composite convex functions, where these functions are the sum of a convex function with Lipschitz-

continuous gradient and some simple convex functions for which we can easily compute their proximal operator. Davis and Yin [2015] also propose the three-operator splitting scheme, which is a similar approach to minimise composite convex functions of the form $f + g + h$, where f is a convex smooth function, g and h are both convex and associated to easy computed proximal operator, but for which the proximal operator of $f + g$ is not easily computable. The algorithm we used is proposed as Algorithm 7. This algorithm is designed to be used with deterministic proximal gradient, though we used it with the stochastic version by estimating the gradient $\nabla\ell(\Omega)$ in the same way as in Algorithm 6 and by forcing the gradient steps γ_t to be non-increasing. Note that applying the compact constraint does not insure that Δ_t will remain in K_ρ for every iteration, it only tends to bring Δ_t back inside K_ρ. In practice, to guarantee that Δ_t remains definite positive, one must control the gradient step γ_n.

Complexity of the stochastic proximal gradient Observe that the left term $\frac{1}{M}\sum_{j=1}^{M} F^{(j)}$ in (3.23) is independent of the parameters Ω and can thus be calculated once and for all at the beginning of the algorithm. Considering this, the complexity of each iteration is $\mathcal{O}(m_t)(|\mathcal{C}|^2 + |\mathcal{Q}|^2 + |C||Q|)$, where $|\mathcal{C}|$ and $|\mathcal{Q}|$ denotes respectively the number of categorical and quantitative variables. The complexity broadly comes from the Wolff sampling Algorithm 1 and the Algorithm 5 for estimating $\nabla \log Z_\Omega$, where mainly basic matrix operations have to be done for every sample of the simulated Markov Chain.

Number of Markov chain runs and gradient step size The learning parameters $\{\gamma_t\}$ and $\{\eta_t\}$ have to be chosen carefully. When $\nabla\ell$ is L-Lipschitz, the proximal gradient algorithm is known to converge with rate $\mathcal{O}(1/t)$ when a fixed step size $\gamma_t = \gamma \in [0, 1/L[$ is used (Parikh and Boyd [2013]). However in practice the Lipschitz constant is unknown and γ_t can be determined at each iteration using a line search algorithm (e.g., Algorithm 4).

The choice of the length η_t of the simulated Markov chain at the t-th iteration depends on the bias of the estimator H of $\nabla\ell$, defined in equation (3.23). Namely, Atchade et al. [2015] show that a constant Markov chain length $\eta_t = \eta$ can be used in the case of an unbiased estimator. However, even if $H_\eta(\Omega)$ is asymptotically unbiased, i.e., $\mathbb{E}[H_\eta(\Omega)] - \nabla\ell(\Omega) \to 0$ for $\eta \to \infty$, the estimator is biased, i.e., $\mathbb{E}[H_\eta(\Omega)] \neq \nabla\ell(\Omega)$ for all $\eta > 0$. As a consequence, the Markov chain length $\{\eta_t\}$ can not be constant and has to increase as a function of the number of iterations. Atchade et al. [2015] show that, when the step sizes $\{\gamma_t\}$ is constant, it is optimal to consider a Markov chain length increasing linearly with the iterations. The same discussion also shows that, when the gradient step size is vanishing, the convergence of the proximal gradient is optimal when η_t increases as $t\gamma_t$, but yet yield a slower convergence rate and a higher computational cost.

Choosing the hyperparameters λ Finally, the hyperparameters $\lambda = (\lambda_\Theta, \lambda_\Delta, \lambda_\Phi)$ can be fixed using cross-validation or using a model selection criterion. Lee and Hastie [2015] treat all the three parameters $\lambda_\Theta, \lambda_\Delta$ and λ_Φ as equal, but also propose a calibration where the regularisers are weighted.

3.2.5 Learning a mixed model using the pseudo-likelihood

In this section, we address the learning task by optimising a penalised pseudo-likelihood instead of using the classic likelihood.

3.2.5.1 Definition of the pseudo-likelihood

Pseudo-likelihood

In this section, we introduce the *pseudo-likelihood* as an alternative to the classic likelihood for estimating the parameters Ω. The pseudo-likelihood approach has been introduced by Besag [1975], and relies on a computationally efficient and consistent estimator.

Definition 3.2 Pseudo-likelihood Given a dataset $\mathcal{D} = \{X^{(j)}\}_{j=1\dots M}$ of M samples and a model Ω, the pseudo log-likelihood is defined by

$$p\ell(\Omega : \mathcal{D}) = \frac{1}{M}\sum_{j=1}^{M} \log p_\Omega(X_{\mathcal{Q}}^{(j)} | X_{\mathcal{C}}^{(j)}) + \frac{1}{M}\sum_{j=1}^{M}\sum_{i\in\mathcal{C}} \log p_\Omega(X_i^{(j)} | X_{-i}^{(j)}), \tag{3.25}$$

where X_{-i} denotes all the rest of the variables except X_i.

Note that our definition of the pseudo-likelihood is slightly different that the usual one

$$\frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{n} \log p_\Omega(X_i^{(j)} | X_{-i}^{(j)}), \tag{3.26}$$

since we distinguish the quantitative and categorical variables, and treat the quantitative part as a conditional Gaussian likelihood given the categorical variables only. In that sense, our pseudo-likelihood is closer from the classic likelihood than the one proposed by Besag [1975] and studied in the context of graphical model learning by Lee and Hastie [2015]. Note also that this pseudo-likelihood is defined over a parametrization space such that Θ and Δ are, respectively, symmetric and symmetric positive-definite.

Proposition 3.5 *The log-pseudo-likelihood* (3.25) *is a concave function of the parameters* Ω.

The log-pseudo-likelihood (3.25) decomposes as a sum of two terms, the left one, associated to the conditional probability of the quantitative variables given the categorical variables, and

the right one, associated to the conditional probability of each categorical variable given all the remaining variables.

Lets focus on the left term. By the proposition 3.1.i, we know that given the categorical variables $x_{\mathcal{C}}$, the quantitative variables have a conditional Gaussian likelihood with mean $\Delta^{-1}(\mu + \Phi^T x_{\mathcal{C}})$ and variance Δ^{-1}. We define $\upsilon = \mu + \Phi^T x_{\mathcal{C}}$, and we want to show that the function

$$\Psi_x : \begin{array}{c} \mathbb{R}^{|\mathcal{Q}|} \times \mathbb{R}^{|\mathcal{Q}|\times|\mathcal{Q}|} \to \mathbb{R} \\ (\upsilon, \Delta) \mapsto \log p_{\mathcal{N}(\Delta^{-1}\upsilon, \Delta^{-1})}(x) \end{array}$$

is concave in $(\upsilon,\ \Delta)$, for every $x \in \mathbb{R}^{|\mathcal{Q}|}$. We have that

$$\begin{aligned} \Psi_x(\upsilon, \Delta) &= -\frac{1}{2}\log \text{Det}(\Delta^{-1}) - \frac{|\mathcal{Q}|}{2}\log(2\pi) - \frac{1}{2}\left(x - \Delta^{-1}\upsilon\right)^T \Delta \left(x - \Delta^{-1}\upsilon\right) \\ &= \frac{1}{2}\log \text{Det}(\Delta) - \frac{|\mathcal{Q}|}{2}\log(2\pi) - \frac{1}{2}x^T\Delta x + \upsilon^T x - \upsilon^T\Delta^{-1}\upsilon. \end{aligned}$$

The term $-\frac{1}{2}x^T\Delta x + \upsilon^T x$ is linear in υ and Δ. The term $\log \text{Det}(\Delta^{-1})$ is concave in Δ (see [Boyd and Vandenberghe, 2004, Section A.4.1]). Let's analyse the last term $\tilde{\Psi}(\upsilon, \Delta) = -\upsilon^T\Delta^{-1}\upsilon$. Since Δ is symmetric, we get the gradients

$$\begin{aligned} \nabla_\upsilon \tilde{\Psi}(\upsilon, \Delta) &= -2\Delta^{-1}\upsilon, \\ \nabla_\Delta \tilde{\Psi}(\upsilon, \Delta) &= \Delta^{-1}\upsilon\upsilon^T\Delta^{-1}. \end{aligned}$$

We denote $g(\upsilon, \Delta) = \left(\nabla_\upsilon \tilde{\Psi}(\upsilon, \Delta), \nabla_\Delta \tilde{\Psi}(\upsilon, \Delta)\right)^T$. Then its differential $\mathrm{d}g_{|\upsilon,\Delta}$ is defined by

$$\mathrm{d}g_{|\upsilon,\Delta} : \begin{array}{c} \mathbb{R}^{|\mathcal{Q}|} \times \mathbb{R}^{|\mathcal{Q}|\times|\mathcal{Q}|} \to \mathbb{R}^{|\mathcal{Q}|} \times \mathbb{R}^{|\mathcal{Q}|\times|\mathcal{Q}|} \\ (h, H) \mapsto \begin{bmatrix} -2\Delta^{-1}h + 2\Delta^{-1}H\Delta^{-1}\upsilon \\ \Delta^{-1}h\upsilon^T\Delta^{-1} + \Delta^{-1}\upsilon h^T\Delta^{-1} - \\ \Delta^{-1}H\Delta^{-1}\upsilon\upsilon^T\Delta^{-1} - \Delta^{-1}\upsilon\upsilon^T\Delta^{-1}H\Delta^{-1} \end{bmatrix}. \end{array}$$

Notice that here, we are using the fact that the differential of the gradient is exactly the Hessian matrix. We now compute $\left\langle \mathrm{d}g_{|\upsilon,\Delta}(h, H), (h, H)\right\rangle$ for any $(h, H) \in \mathbb{R}^{|\mathcal{Q}|} \times \mathbb{R}^{|\mathcal{Q}|\times|\mathcal{Q}|}$, where here $\langle\cdot,\cdot\rangle$ is the dot product defined, for any $h_1, h_2 \in \mathbb{R}^{|\mathcal{Q}|}$ and $H_1, H_2 \in \mathbb{R}^{|\mathcal{Q}|\times|\mathcal{Q}|}$, by

$$\langle (h_1, H_1), (h_2, H_2)\rangle = \text{Tr}\left(H_1 H_2^T\right) + h_1^T h_2,$$

and we now show that it is always negative for any (h, H). We have that

$$\begin{aligned}\left\langle \mathrm{d}g_{|\upsilon,\Delta}(h,H),(h,H)\right\rangle &= -2(H^T\Delta^{-1}\upsilon)^T\Delta^{-1}(H^T\Delta^{-1}\upsilon) + 4h^T\Delta^{-1}(H\Delta^{-1}\upsilon) - 2h^T\Delta^{-1}h \\ &= -(H^T\Delta^{-1}\upsilon + h)^T\Delta^{-1}(H^T\Delta^{-1}\upsilon + h) \leq 0.\end{aligned}$$

Hence, $\tilde{\Psi}$ is a concave function of (υ, Δ). We thus proved that the Gaussian term (left term) in the log-pseudo-likelihood (3.25) is concave in the parameters Δ.

Concerning the Ising part of (3.25), the conditional distribution of x_i with $i \in \mathcal{C}$, given all the other variables x_{-i}, is

$$p_\Omega(x_i \mid x_{-i}) \propto \exp\left[\theta_{ii}x_i^2 + x_i\left(\sum_{j>i}\theta_{ij}x_j + \sum_{u\in\mathcal{Q}}\phi_{iu}x_u\right)\right].$$

Each variable $x_i, i \in \mathcal{C}$ has a conditional Bernoulli distribution with mean

$$\mathbb{E}_\Omega[x_i \mid x_{-i}] = \frac{\mathrm{e}^{q_\Omega(x,i)}}{1+\mathrm{e}^{q_\Omega(x,i)}},$$

with

$$q_\Omega(x, i_0) = \theta_{i_0 i_0} + 2\Theta_{i_0,-i_0}x_{\mathcal{C},-i_0} + \Phi_{i_0,\mathcal{Q}}x_{\mathcal{Q}}, \tag{3.27}$$

where $x_{\mathcal{C},-i_0}$ denotes the vector $X_\mathcal{C}$ with the i_0 entry removed, Θ_{-i_0,i_0} represents $(\theta_{i_0,j})_{j\neq i_0}$, the i_0th line of Θ without the i_0-th element, and $\Phi_{i_0,\mathcal{Q}}$ represents the i_0th line of Φ. We thus have

$$\begin{aligned}\log p_\Omega(X_{i_0} \mid X_{-i_0}) &= X_{i_0}\log P_\Omega(X_{i_0}=1 \mid X_{-i_0}) \\ &\quad + (1-X_{i_0})\log(1-P_\Omega(X_{i_0}=1 \mid X_{-i_0})) \\ &= X_{i_0}q_\Omega(X,i_0) - \log(1+\exp q_\Omega(X,i_0)).\end{aligned}$$

The left term of this formula is linear in the parameters Ω. Similarly to the Gaussian part, it is easy to show that the right term is concave in the parameters Ω. Finally, as a sum of concave and linear terms, the pseudo-log-likelihood (3.25) is concave.

Surprisingly, the maximum pseudo-likelihood estimator approach is consistent and yields an exact solution of in the case where the data is generated by a model p_{Ω^*} and when $M \to \infty$ where M is the number of samples in the training dataset. The strong consistency of the (classic) pseudo-likelihood (3.26) has been proved in the case of Ising model by Guyon and Künsch [1992] and in the case of continuous models by Mase [1995].

3.2.5.2 Optimising the pseudo-likelihood

We now focus on the calculation of the derivative of the pseudo log-likelihood. Concerning the Gaussian left part of the pseudo log-likelihood (3.25), by the property 3.1.i, given $X_{\mathcal{C}}$, $X_{\mathcal{Q}}$ admits a conditional normal density with mean $\Delta^{-1}(\mu + \Phi^T X_{\mathcal{C}})$ and covariance matrix Δ^{-1}. We thus have

$$\begin{aligned}\log p_\Omega(X_{\mathcal{Q}} \mid X_{\mathcal{C}}) = &-\frac{1}{2} X_{\mathcal{Q}}^T \Delta X_{\mathcal{Q}} + (\mu + \Phi^T X_{\mathcal{C}})^T X_{\mathcal{Q}} \\ &- \frac{1}{2}(\mu + \Phi^T X_{\mathcal{C}})^T \Delta^{-1}(\mu + \Phi^T X_{\mathcal{C}}) \\ &+ \log[(2\pi)^{-\frac{|\mathcal{Q}|}{2}} |\Delta|^{\frac{1}{2}}].\end{aligned}$$

Differentiating in Δ, Φ and μ yields the gradients

$$\begin{aligned}\nabla_\Delta \log p_\Omega(X_{\mathcal{Q}} \mid X_{\mathcal{C}}) &= \frac{1}{2}\left[-X_{\mathcal{Q}} X_{\mathcal{Q}}^T + \Delta^{-1} + \Delta^{-1}(\mu + \Phi^T X_{\mathcal{C}})(\mu + \Phi^T X_{\mathcal{C}})^T \Delta^{-1}\right], \\ \nabla_\Phi \log p_\Omega(X_{\mathcal{Q}} \mid X_{\mathcal{C}}) &= X_{\mathcal{C}} X_{\mathcal{Q}}^T - X_{\mathcal{C}}(\mu + \Phi^T X_{\mathcal{C}})^T \Delta^{-1}, \\ \nabla_\mu \log p_\Omega(X_{\mathcal{Q}} \mid X_{\mathcal{C}}) &= X_{\mathcal{Q}} - \Delta^{-1}\mu - \Delta^{-1}\Phi^T X_{\mathcal{C}}.\end{aligned} \tag{3.28}$$

Concerning the Ising right part of the pseudo log-likelihood (3.25), observe that, by (3.27), we have the gradients, for all $i, j \in \mathcal{C}$,

$$\nabla_{\Theta_{i,j}} q_\Omega(X, i_0) = \begin{cases} \mathbb{1}_{\{i=i_0\}} & \text{if } i = j, \\ \mathbb{1}_{\{i=i_0\}} X_j + \mathbb{1}_{\{j=i_0\}} X_i & \text{if } i \neq j. \end{cases}$$

and, for all $i \in \mathcal{C}$ and $v \in \mathcal{Q}$,

$$\nabla_{\Phi_{i,v}} q_\Omega(X, i_0) = \mathbb{1}_{\{i=i_0\}} X_v.$$

It follows that

$$\begin{aligned}\nabla_\Theta \sum_{i_0 \in \mathcal{C}} \log p_\Omega(X_{i_0} \mid X_{-i_0}) = &\operatorname{Diag}\left(E_\Omega(X) \circ (2X_{\mathcal{C}} - 1)\right) \\ &- \operatorname{Diag}\left(-X_{\mathcal{C}}\right) + 2X_{\mathcal{C}} X_{\mathcal{C}}^T - (E_\Omega(X) X_{\mathcal{C}}^T + X_{\mathcal{C}} E_\Omega(X)^T),\end{aligned} \tag{3.29}$$

where here $\operatorname{Diag}(A)$ denotes the diagonal matrix with diagonal A, $A \circ B$ denotes the Hadamard product of two matrices A and B, and $E_\Omega(X)$ is the vector defined of size $|\mathcal{C}|$, whose i-th com-

ponent is given by

$$E_\Omega(X)_i = p_\Omega(X_i = 1 \mid X_{-i}) = \frac{\mathrm{e}^{q_\Omega(X,i)}}{1 + \mathrm{e}^{q_\Omega(X,i)}}, \quad i \in \mathcal{C}.$$

Similarly, we get that

$$\nabla_\Phi \sum_{i_0 \in \mathcal{C}} \log P_\Omega(X_{i_0} \mid X_{-i_0}) = X_\mathcal{C} X_\mathcal{Q}^T - E_\Omega(X) X_\mathcal{Q}^T. \tag{3.30}$$

Note also that another way to write $q_\Omega(X, \mathcal{C}) = (q_\Omega(X, i))_{i \in \mathcal{C}}$ is to set

$$q_\Omega(X, \mathcal{C}) = (\Theta + \Theta^T) X_\mathcal{C} + \mathrm{Diag}\,(\Theta) \circ (1 - 2X_\mathcal{C}) + \Phi X_\mathcal{Q}.$$

Those equations carry out an algorithm for structure learning, using proximal gradient. Contrary to the classic likelihood (3.19), all terms of the pseudo-likelihood can be expressed in closed-form. Thus we use the deterministic proximal gradient Algorithm 3: if Ω_0 denotes the starting estimates, and $\{\gamma_t\}$ a sequence of positive step sizes, then given $\Omega^{(t)}$, we compute

$$\Omega^{(t+1)} = \mathrm{Prox}_{\gamma_{t+1} g}(\Omega^{(t)} + \gamma_{t+1} \nabla p\ell(\Omega^{(t)}), \tag{3.31}$$

where $\mathrm{Prox}_{\gamma_t g}$ is the proximal operator defined in (3.13).

Choosing the gradient step γ_t As discussed by Combettes and Pesquet [2011], the proximal gradient algorithm converges with rate $\mathcal{O}(1/t)$ for a gradient-Lipschitz objective function with Lipschitz constant L and fixed gradient step sizes $\gamma_t = \gamma \in [0, 1/L[$. Observe that the pseudo log-likelihood (3.25) is actually gradient-Lipschitz, as a sum of conditional likelihood which are all individually gradient-Lipschitz, yet with different Lipschitz constants. As discussed in section 3.2.4, since the Lipschitz constant is not known, the gradient step sizes γ_t can be determined by line search, for which the Algorithm 4 can be applied.

Complexity of the pseudo-likelihood optimisation The pseudo-likelihood optimisation gives rise to a different complexity than the likelihood optimisation with the stochastic proximal algorithm. Namely, the algorithm still boils down to basic matrix operations as multiplication or inverse calculations, and most terms in the calculation of the gradients (3.28) do not depend on the parameters $\Omega^{(t)}$ and can be done once and for all before starting the iterations of the learning algorithm. However, contrary to the stochastic approach, the calculation of the pseudo-likelihood and its gradient still require a loop over the data in equations (3.29) and (3.30) for computing the terms involving $\sum_{j=1}^m E_\Omega(X^{(t)}, \mathcal{C}) X_\mathcal{Q}^{(t)T}$. The complexity of an iteration of the proximal gradient algorithm is thus $\mathcal{O}(M(|\mathcal{C}|^2 + |\mathcal{C}||\mathcal{Q}|) + |\mathcal{Q}|^2)$.

3.3 Experiments on synthetic data

In this section, we will show some experiments made with synthetic data. The data is instances of an underlying mixed graphical model Ω^* (supposed unknown for the learning) introduced by Lee and Hastie [2015] that uses ten binary variables $x_1, \ldots, x_{10}$ and ten quantitative variables $x_{11}, \ldots, x_{20}$. The structure of the model is presented in Figure 3.4.

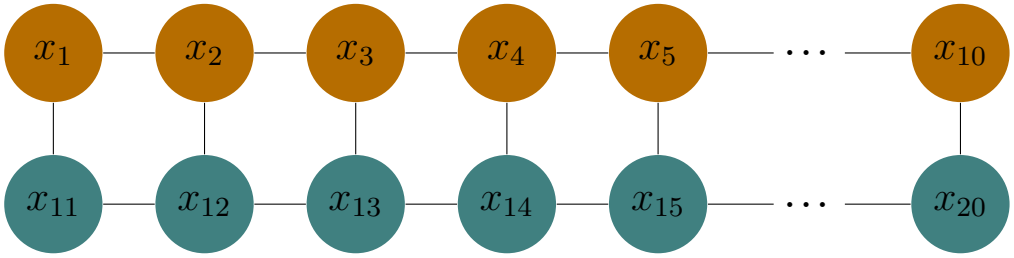


Figure 3.4: Structure of the network used for the experiments. This networks uses ten binary variables $x_1, \ldots, x_{10}$ (in brown, on the top row) and ten quantitative variables $x_{11}, \ldots, x_{20}$ (in grey, on the bottom row). The network has the structure of the a ladder, where each binary node is connected to a unique quantitative node and reciprocally, and the nodes of each type are forming a chain, from x_1 to x_{10} for the binary nodes, and from x_{11} to x_{20} for the quantitative nodes. Here are only represented half of the node for more visibility.

3.3.1 Presentation of the synthetic model

The parameters $\Omega^* = (\Theta^*, \mu^*, \Delta^*, \Phi^*)$ of the model are chosen this way:

- Θ^* is a 10×10 square matrix, with -0.5 for the diagonal entries and 0.5 for the entries of the upper and lower diagonals,
- μ^* is a vector of 10 null entries,
- Δ^* is a 10×10 square matrix with 1 for the diagonal entries and 0.25 for the entries of the upper and lower diagonals,
- Φ^* is a 10×10 square matrix with 0.5 for the entries of the diagonal and 0 elsewhere.

$$\Theta^* = \begin{bmatrix} -0.5 & 0.5 & 0 & \ldots & 0 \\ 0.5 & -0.5 & \ddots & & \\ 0 & \ddots & \ddots & & \\ \vdots & & & \ddots & 0.5 \\ 0 & & & 0.5 & -0.5 \end{bmatrix}, \qquad \mu^* = \begin{bmatrix} 0 \\ \vdots \\ \\ \vdots \\ 0 \end{bmatrix},$$

$$\Delta^* = \begin{bmatrix} 1 & 0.25 & 0 & \dots & 0 \\ 0.25 & 1 & \ddots & & \\ 0 & \ddots & \ddots & & \\ \vdots & & & \ddots & 0.25 \\ 0 & & & 0.25 & 1 \end{bmatrix}, \quad \Phi^* = \begin{bmatrix} .5 & 0 & \dots & \\ 0 & .5 & & \\ \vdots & & \ddots & 0 \\ 0 & & 0 & .5 \end{bmatrix}.$$

For sampling the data, we use the sampling Algorithm 2. For each sampled Markov chain, we discard the first 20 samples and we then keep only one sample every 20 samples.

3.3.2 Structure recovery experiments

The Figure 3.5 shows the results of experiments on structure recovery. We compare our methods to several structure learning algorithms:

1. The stochastic proximal gradient algorithm presented in section 3.2.4.
2. The pseudo-likelihood approach we presented in section 3.2.5.
3. The BayesiaLab [2013] software, which can learn Bayesian network only from categorical data. The quantitative variables have been categorised by the software, and the structure is learned by optimising a BIC score.
4. The mixed graphical model (mgm) R package developed by Haslbeck and Waldorp [2015a] that uses ℓ_1-regularised neighbourhood regressions. We have used the standard parameters proposed by the package, and in particular, we used the Extended Bayesian Information Criterion procedure to select the parameter for the Lasso penalisation, and we used two degrees of augmented interactions.
5. The maximum pseudo-likelihood approach of Lee and Hastie [2015], where the regularised pseudo-likelihood is optimised using deterministic proximal gradient. We reused the Matlab codes available on Jason Lee's webpage (using the UGM and TFOCS Matlab framework, see Becker et al. [2011]).
6. The shrinkage covariance algorithm, which is the estimator, for $\alpha \in [0, 1]$,

$$\Sigma_{\text{shrunk}} = (1 - \alpha)\hat{\Sigma} + \alpha \frac{\text{Tr}\left(\hat{\Sigma}\right)}{n} \text{Id},$$

where $\hat{\Sigma}$ is the empirical covariance and Id is the identity matrix.

In real applications, the underlying model Ω^* is unknown and the choice of $\lambda = (\lambda_\Theta, \lambda_\Delta, \lambda_\Phi)$ require a significant amount of computing by using a model selection criterion or a cross-validation procedure. For our approaches with stochastic proximal gradient and pseudo-likelihood optimisation, we have set the regularisation parameters $\lambda_\Theta, \lambda_\Delta$ and λ_Φ proportional to $\sqrt{\log(n)/M}$ for all the simulations (as suggested by Wainwright et al. [2006]). We chose the proportionality constants by trials and errors with a dataset containing 3000 samples, i.e., they are the values for which the zeros of Θ, Δ and ϕ are correctly recovered when there is enough data. Note that these constants are different for each regulariser. Namely, for the pseudo-likelihood approach we used

$$\lambda_\Theta = 1.4\sqrt{\frac{\log(20)}{M}} \qquad \lambda_\Delta = 0.95\sqrt{\frac{\log(20)}{M}} \qquad \lambda_\Phi = 4.6\sqrt{\frac{\log(20)}{M}}, \tag{3.32}$$

and for the stochastic proximal gradient approach we used

$$\lambda_\Theta = 3.7\sqrt{\frac{\log(20)}{M}} \qquad \lambda_\Delta = 3.2\sqrt{\frac{\log(20)}{M}} \qquad \lambda_\Phi = 4.3\sqrt{\frac{\log(20)}{M}}. \tag{3.33}$$

Following our discussion in section 3.2.3 and in section 3.2.5.2, we have chosen a fixed gradient step size $\gamma = 1$ for the pseudo-likelihood approach and $\gamma = 0.1$ for the stochastic approach.

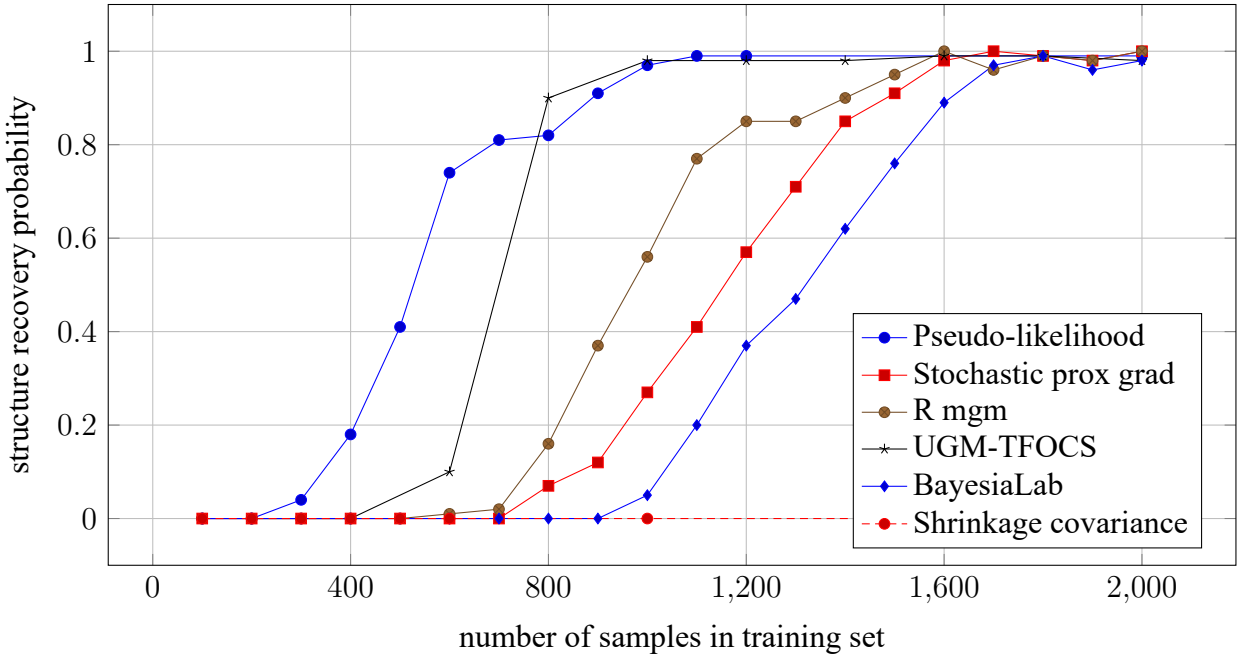


Figure 3.5: Probability of recovering the true network structure, i.e., every true edge from Ω^* is selected and no false edge – absent in Ω^* – is selected, for a given sample size. Remember that Ω^* is a network with 10 categorical and 10 quantitative variables, and its structure is presented in Figure 3.4. Each training set is i.i.d. sampled from p_{Ω^*} using Algorithm 2. Each displayed point is the average of 100 trials.

Note that distinguishing the three regularisers instead of choosing the same values yields a

better structure recovery. The figure 3.6 shows that choosing the regularisers separately leads to better structure recovery probability with less samples.

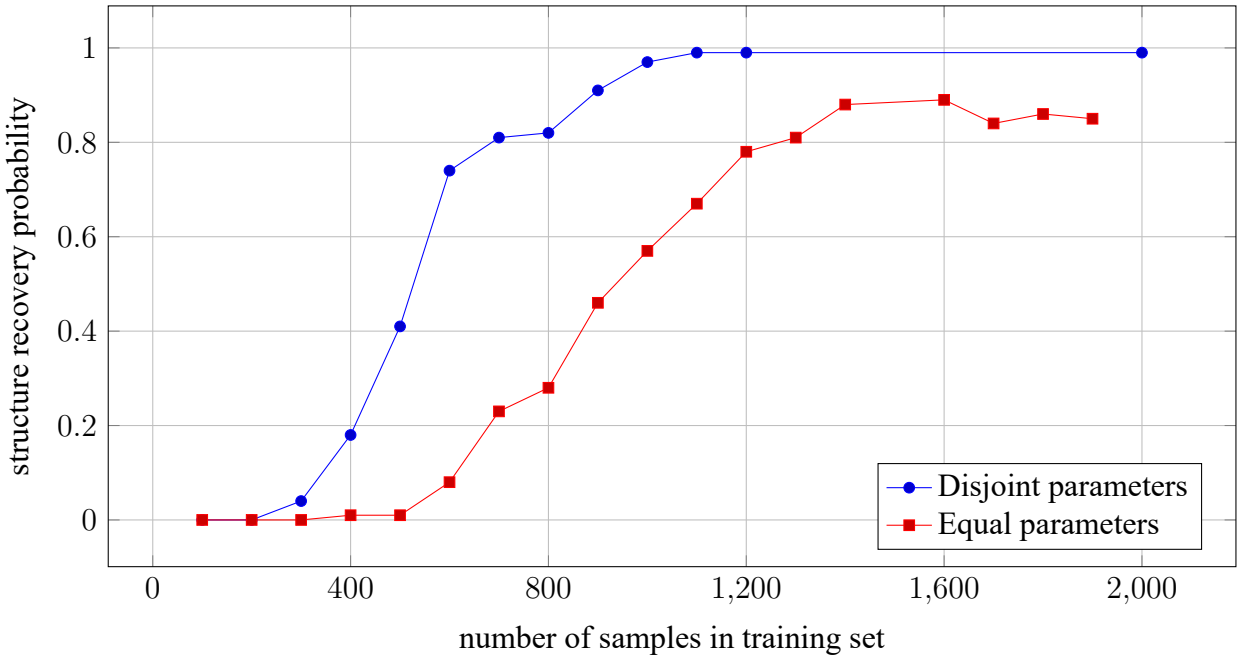


Figure 3.6: Comparison of the probability of structure recovery between the cases where the regularisers are equals and whey they are chosen separately. The results use for the disjoint case are the same than the ones in Figure 3.5 for our pseudo-likelihood approach. We can clearly see that taking all the three regalirisers $\lambda_\Theta, \lambda_\Delta$ and λ_Φ leads to a poorer structure recovery algorithm.

Concerning the Bayesian network learning, the structure of a Bayesian network and a Markov network are not directly comparable. However, we can still compare the set of conditional independences encoded by each graph. Generally speaking, it not possible to find a oriented structure that encodes the same set of independences encoded by an undirected structure, and reciprocally. In our case, the set of conditional independences in the graph represented on the Figure 3.4 is defined by the notion of separation (introduced in section 2.4.5.1). For example, if we look at the four nodes x_1, x_2, x_{11}, x_{12} on the Figure 3.4, we see that the independences $x_1 \perp x_{12} \mid x_2, x_{11}$ and $x_2 \perp x_{11} \mid x_1, x_{12}$ hold for this network. However, there is no oriented structure for which this two independences holds together (a similar case is studied by [Friedman and Koller, 2009, Chapter 4]). Hence there is no Bayesian structure that encodes the whole set of conditional independences of p_{Ω^*}. More generally, observe that each square subgraph $x_i - x_{i+1} - x_{i+11} - x_{i+10} - x_i$ on the Figure 3.4 defines two conditional independences $x_i \perp x_{i+11} \mid x_{i+1}, x_{i+10}$ and $x_{i+1} \perp x_{i+10} \mid x_i, x_{i+11}$, and only one of them can be encoded by a single Bayesian structure. Any Bayesian structure learned with enough data sampled from p_{Ω^*} will encode only one of the two conditional independences defined by every square subgraph of the Figure 3.4. Reciprocally, any conditional independence of the form $x \perp y|z$ encoded by

the learned Bayesian network holds in p_{Ω^*} and is thus also encoded by the mixed undirected network. We thus state that a Bayesian network has correctly recovered the structure of the undirected network parametrised by Ω^* when all the conditional independences of the form $x \perp y|z$ that is encoded by the Bayesian network is also encoded by the Markov network, and when only one of the two conditional independences encoded by each square subgraph in the Markov network is also encoded in the Bayesian network. Note that in the case of the ladder-shaped structure of Figure 3.4, it means that the skeleton of the learned Bayesian network (i.e., when every oriented edge of the Bayesian network has been replaced by an undirected edge) is exactly the structure of the network in Figure 3.4.

The results of the structure recovery probability estimation is presented in Figure 3.5. We first notice that the shrinkage covariance approach (based on the calculation of the empirical covariance) never recovers the true structure, as highlighted by Loh et al. [2013]. Secondly, we see that our pseudo-likelihood approach slightly outperforms the pseudo-likelihood approach of Lee and Hastie [2015].

We can also observe that the structure of the graph is learned relatively fast, i.e., it does not evolve after a few iterations, especially for the pseudo-likelihood approach. This result is known for the Lasso, see Liang et al. [2014]. The Figure 3.7 shows the evolution of the True Positive Rate (TPR) and False Discovery Rate (FDR) during the learning of two graphs, one with 20 variables and one with 200 variables. The TPR and FDR are respectively the rate of recovered edges that exist in Ω^* and the rate of edges that do not exist in Ω^*. They are defined as

$$\text{TPR}_t = \frac{\sum_{i<j} \mathbb{1}_{\{|\omega_{ij}^{(t)}|>0\}} \mathbb{1}_{\{|\omega_{ij}^*|>0\}}}{\sum_{i<j} \mathbb{1}_{\{|\omega_{ij}^*|>0\}}}, \quad \text{FDR}_t = \frac{\sum_{i<j} \mathbb{1}_{\{|\omega_{ij}^{(t)}|>0\}} \mathbb{1}_{\{\omega_{ij}^*=0\}}}{\sum_{i<j} \mathbb{1}_{\{|\omega_{ij}^{(t)}|>0\}}},$$

where Ω is here seen as a symmetric matrix defined by blocs by

$$\Omega = \left(\begin{array}{c|c} \Theta & \Phi \\ \hline \Phi^T & \Delta \end{array}\right).$$

Observe that as shown in Figure 3.5, the maximum pseudo-likelihood estimator correctly recovers the structure, even in high dimension, whereas the (stochastic) maximum likelihood estimator performs poorly. There is no notable difference between the low and high dimension for the maximum pseudo-likelihood estimator, since with the ladder structure of Ω^*, each node has a fixed number of neighbours - 3 in general, and 2 for the extremities of the structure - and hence the conditional distributions of each node have few parameters and are easy to learn. However, as the number of parameters of the conditional distributions increases, the pseudo-likelihood approach defaces for non-sparse structures.

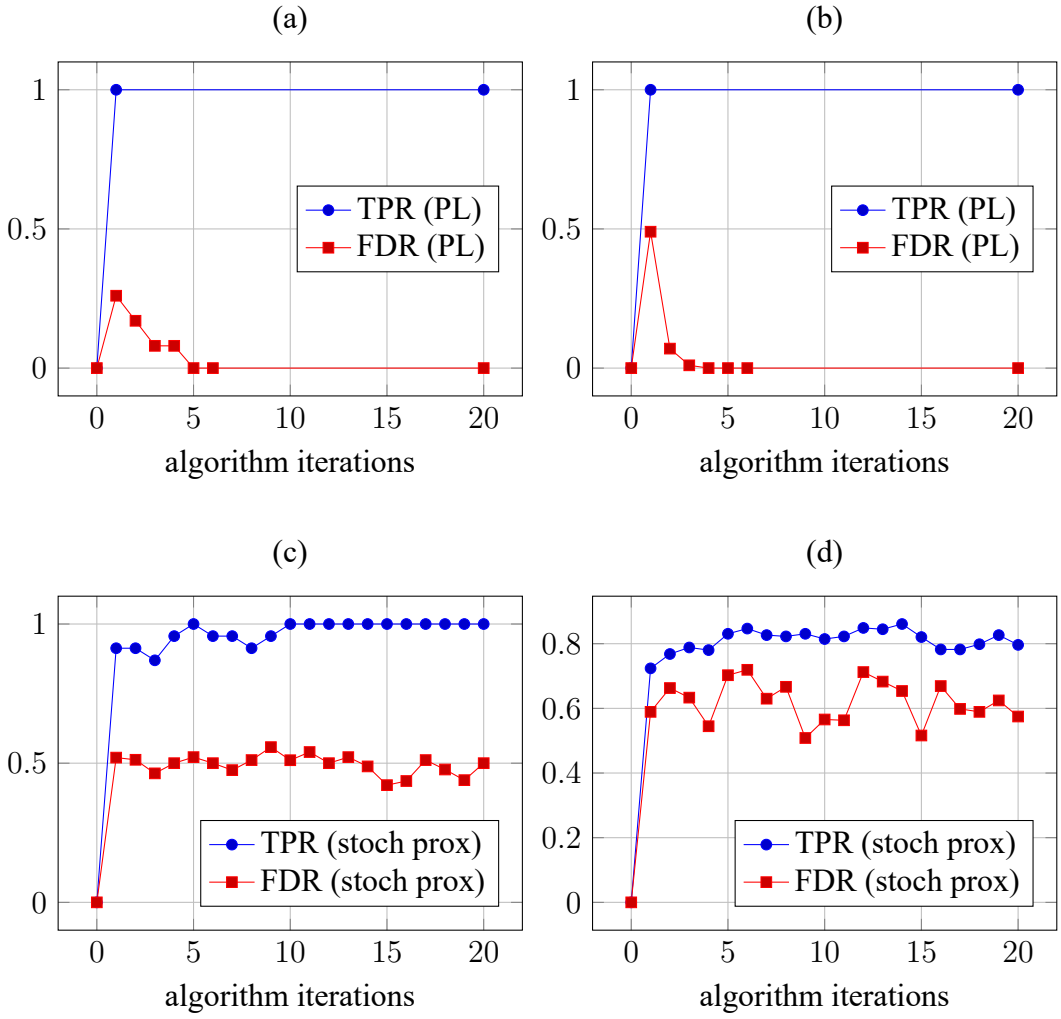


Figure 3.7: Evolution of True Positive Rates and False Discovery Rate during the learning of a mixed graphical model. The data have been sampled from a mixed graphical model with a ladder shape, similar to the structure of Figure 3.4, with 10 categorical and 10 quantitative variables for the graphs (a) and (c), and 100 categorical and 100 quantitative variables for the graphs (b) and (d). For all graphs, we used 1000 samples in the training set.

We end this experiment section by comparing the execution times. For both stochastic proximal gradient and pseudo-likelihood approaches, we compare the execution time and the number of iterations required to reach a fixed score change between two successive score measures. Note that for the stochastic approach, since we estimate the log-likelihood with Markov chain simulations, the score is never strictly decreasing but has a decreasing tendency. We decided to stop the learning when the average of score computed over a small time windows is not varying enough, i.e., when the difference between two successive averaged score is lower than a fixed threshold. The Figure 3.8 illustrates the experiment done by varying the samples sizes of the training datasets with a fixed dimension of 10 categorical and 10 quantitative variables, and the Figure 3.9 shows the results of an experiment made with fixed sample sizes and by varying the dimension. As expected, the execution time of an iteration of the stochastic proximal gradient does not depend on the sample size.

M	stochastic prox		pseudo-likelihood	
	mean execution time (sec)	iterations	execution time (sec)	iterations
100	3.34	274	0.12	445
1000	3.32	265	0.51	507
10000	3.45	287	0.59	495
100000	3.43	279	6.25	603
1000000	3.48	294	67	576

Figure 3.8: Execution time of each iteration of the stochastic proximal gradient learning Algorithm 3 and the pseudo-likelihood optimisation Algorithm 3, when the dimension is fixed to $|\mathcal{C}| = 100$ and $|\mathcal{Q}| = 100$, for various sample sizes M.For the pseudo-likelihood approach, we stopped the learning when the difference between two successive scores was less than 0.0001. For the MCMC approach, we stopped when the difference between two averaged scores is less than 0.1.

$\lvert\mathcal{C}\rvert + \lvert\mathcal{Q}\rvert$	stochastic prox		pseudo-likelihood	
	mean execution time (sec)	iterations	execution time (sec)	iterations
20	.76	82	0.005	151
100	1.21	388	0.12	445
200	3.3	412	0.50	626
1000	42	1015	11	1634
2000	94	1867	59	2040

Figure 3.9: Execution time of each iteration of the stochastic proximal gradient learning Algorithm 3 and the pseudo-likelihood optimisation Algorithm 3, when the sample size is fixed to $M = 1000$ and the dimension $n = |\mathcal{C}| + |\mathcal{Q}|$ is varying. For the pseudo-likelihood approach, we stopped the learning when the difference between two successive scores was less than 0.0001. For the MCMC approach, we stopped when the difference between two averaged scores is less than 0.1.

Chapter 4

Anomaly Localisation

Anomaly detection (Chandola et al. [2009]) refers to the task of finding unusual elements in a set of observations. Most of the existing works on anomaly detection are focusing on point (unconditional) anomaly detection (see section 2.4.1) and are looking for outliers with respect to all the features in the dataset. In this section, we rather study the *conditional anomaly detection* problem (Chandola et al. [2009], Valko [2011]), i.e., the problem of detecting unusual values in a subset of variables given the values of the remaining variables.

Conditional Anomaly Detection

In the industrial terminology, the conditional anomaly detection task is referred as the *localisation problem*. The data is often collected by sensors and the instanced variables often correspond to physical measures or states of specific parts and components of the systems, and localising which part of the system is failing is a valuable information. However, for complex system, the number of variables can be very high so that finding the cause(s) of a detected anomaly is unachievable, even for an expert team. In particular, the wave guide case explained in the Example 2.2 depicts a case where localising the causes of unexpected displays on the pilot screen has required months of investigation by an expert team. For this specific case, the benefits of using Bayesian networks has been revealed by Kemkemian et al. [2013], where a Bayesian network has been learned with data produced by a radar behaving normally, and where the root causes of the anomaly were localised by manually analysing the conditional probability distributions of each of the 140 variables of the problem.

Localisation problem

In this section, we present our approach for solving the localisation task in the industrial context of RBE2 production in Thales.

4.1 Definition of the localisation task

The concept of conditional anomaly in data is somewhat ambiguous in the literature. Several definitions have been proposed in the past (Markou and Singh [2003a], Markou and Singh

[2003b], Valko et al. [2011]), and different detection approaches have been proposed (see section 2.4.1). In particular, Valko [2011] study the case of detecting unusual values of one variable given the values of the remaining variables, where the variables are all categorical and take a finite number of values. Namely, given a dataset $\{X^m\}_{m=1...M}$ of M samples, a conditional anomaly in the value $x_i^{(m)}$ in the m-th sample given the remaining values $x_{-i}^{(m)}$ is detected when $p(x_i^{(m)} \mid x_{-i}^{(m)}) = \mathcal{O}(\exp(-\kappa M))$ for some level κ, where p is the underlying density of the data. Several approaches addressing the inaccessibility of this underlying model are discussed.

Concerning our study, the conditional anomalies can not be defined in term of single samples. From an industrial perspective, breakdowns might be generated by isolated samples behaving anomalously, but they only encompass a small fraction of all possible anomalies, like anomalous accumulations or drifts. From a statistical perspective, most of the measures are made with sensors which are subject to noise or easily disturbed, and focusing only on the outliers might leads to a high false alarm rate.

In our study, we will define the conditional anomalies in term of change in the parameters of the conditional probability distributions of the variables. The Figure 4.1 and Figure 4.2 depict a synthetic case, close to potential real situations, for which the anomalies lie in a small change of the parameters of the underlying density sampling the data. On both figures, data were drawn from two univariate Gaussian distributions, representing a normal and an anomalous situation. If some outliers can be visually identified with a threshold approach, a lot of samples from both class might be wrongly labelled.

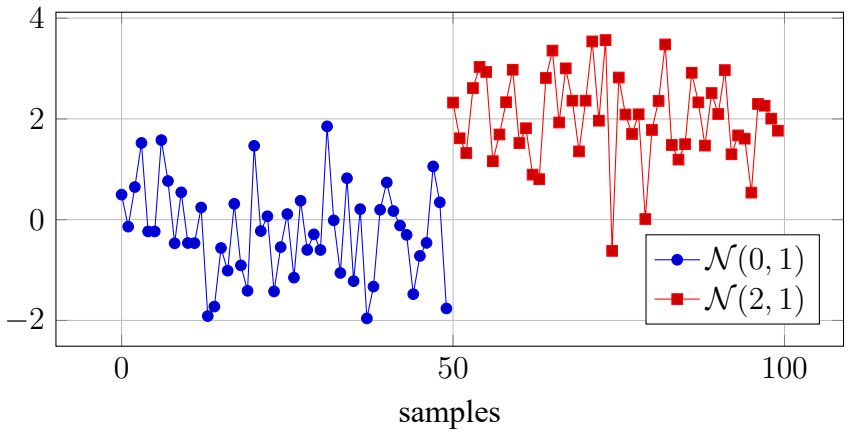


Figure 4.1: Samples of univariate Gaussian distributions. The first fifty samples (in blue) are drawn from $\mathcal{N}(0,1)$ and the last fifty samples (in red) from $\mathcal{N}(0.2,1)$.

The change of parameters in the underlying density has a very valuable meaning for the industrial experts. It can be the evidence of the ageing of a component, an unanticipated correlation, a leak, or more. That is why, rather than detecting the outliers in the data, we are more interested in detecting the time at which the parameters of the underlying conditional density

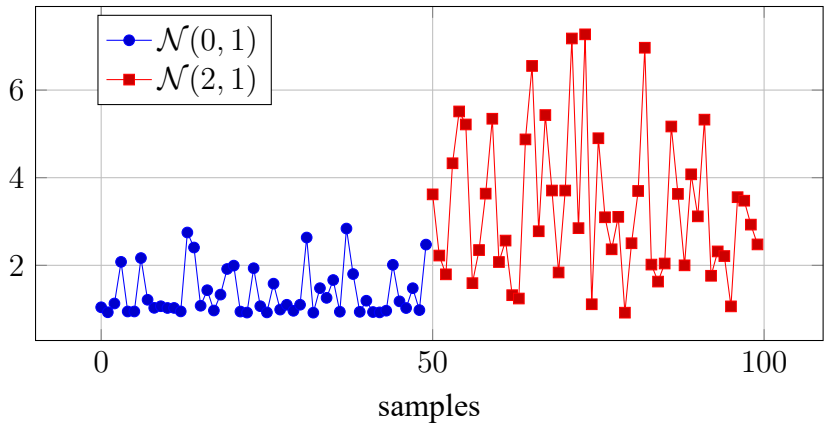


Figure 4.2: Negative log-likelihood of 100 samples calculated with respect to the parameters of the Gaussian distribution $\mathcal{N}(0,1)$. The first fifty samples (in blue) are drawn from $\mathcal{N}(0,1)$ and the last fifty samples (in red) from $\mathcal{N}(0.2,1)$.

have changed and has sampled the outliers. Note that this implies to keep unchanged the time order of the samples.

We have limited our study on changes of the conditional mean of the random variables. Remember that our data is heterogeneous, that is either quantitative or binary. By the proposition 3.1, conditionally to the other variables, each variable has either a Bernoulli distribution, parametrised by its mean value, or a Gaussian distribution, parametrised by a mean and a standard deviation.

These considerations lead to the following definition of the anomaly localisation task. This definition complements the Definition 2.1 proposed in section 2.4.3 by specifying the analysis of the conditional mean of the variables.

Definition 4.1 Anomaly localisation Suppose we have learned a model p_Ω with the parameters Ω over a set of n variables $x_1, \ldots, x_n$, and we have a test set $\mathcal{D} = \{X^{(t)}, t = 1 \ldots M\}$ of M samples, indexed by the time. We define the localisation problem as finding the subset of variables $\{x_i, i \in 1 \ldots n\}$ whose conditional means $\mathbb{E}[X_i^{(t)}|X_{-i}^{(t)}]$, monitored as a function of time have changed compared to the expectation $\mathbb{E}_\Omega[X_i^{(t)}|X_{-i}^{(t)}]$ of the learned model $p_\Omega(X_i^{(t)}|X_{-i}^{(t)})$, where X_{-_i} denotes all the variables except X_i.

4.2 Detection of a change in the conditional mean

In this section, we present our method to detect changes in the conditional mean of random variables. Although we are using batch datasets in our industrial applications (produced by the built-in test during the execution of deterministic scenarios), a future aim of this work is to be embedded in jet fighters for operational use, in which case the data will be produced online. Hence we will focus on online change detection approaches.

The method we propose will simultaneously detect and localise anomalies from a sequence of new data $(X_{\mathcal{C}}^{(t)}, X_{\mathcal{Q}}^{(t)})$, $t = 1, 2, \ldots$ using a reference model p_Ω, where the parameters Ω have already been learned with normal data (see chapter 3). The main idea to localise anomalies is to monitor each term of the log-pseudo-likelihood Besag [1975] as a function of time and detect a change in the parameter of these terms. This problem is commonly referred to as change detection in the statistics literature.

4.2.1 Change detection techniques

Many approaches have been studied to detect a change in the distribution of data, and Basseville et al. [1993] proposes a large survey of online approaches addressing this subject. Namely, given a sequence of i.i.d. observations $\{X^{(t)}\}_t$ with probability distribution $p_\Omega(X)$, we suppose that there is an unknown *change time* t_0 before which the parameters Ω are equal to Ω_0, and after which they are equal to $\Omega_1 \neq \Omega_0$. In the case where the parameters Ω_0 is known, the change-point detection task is to detect the change time t_0 and/or estimate the new parameters Ω_1.

Change time

4.2.1.1 Elementary online parametric change detection algorithms

Most of the change detection techniques rely on the study of the instantaneous *log-likelihood ratio* for a sample X, defined by

Log-likelihood ratio

$$s(X) = \log \frac{p_{\Omega_1}(X)}{p_{\Omega_0}(X)}. \tag{4.1}$$

The following property follows on this definition.

Proposition 4.1 *Let $\mathbb{E}_{\Omega_0}$ and $\mathbb{E}_{\Omega_1}$ denote the expectation respectively under the distributions p_{Ω_0} and p_{Ω_1}, and s denotes the log-likelihood ratio (4.1), then, if p_{Ω_0} and p_{Ω_1} are distinct densities,*

$$\mathbb{E}_{\Omega_0}(s) < 0 \quad \textit{and} \quad \mathbb{E}_{\Omega_1}(s) > 0.$$

This property states that a change in the parameters Ω depicts a change in the sign of the mean of the log-likelihood ratio.

The literature is very wide in change detection, and embrace many situations, whether the samples are independent or not, whether the parameters Ω_0 and Ω_1 are known, or whether the change time t_0 is known. Many of these approaches are exploiting the proposition 4.1, see Basseville et al. [1993] and the references therein for an exhaustive review.

The techniques presented in the following have been developed for the case where Ω_0 and Ω_1 are available.

The Shewhart control charts (see Shewhart [1931], Duncan [1986]) is a tool used in quality control. The samples are divided into set of samples with size M, and for each samples set, the sum of the log-likelihood ratio $S_M = \sum_{t=1}^{M} s_t$ is computed. A change is detected if the sum S_M is greater than a fixed threshold.

The geometric moving average control charts approach, proposed by Roberts [2000], relies on a weighted sum of log-likelihood ratio, to increase the impact of recent samples over old ones. Namely, the decision statistic is $S_k = \sum_{t=0}^{M-1} \gamma_t s_{k-t}$ where the weights γ_t are exponential, i.e., $\gamma_t = \alpha(1-\alpha)^t$ for $0 < \alpha \leq 1$. Again, a change is detected by thresholding the decision function S_k. The filtered derivative approach, introduced by Duncan [1986], uses the same weighted sum of log-likelihood ratio with exponential weights, but the detection of change is made by looking at the differences $\delta S_k = S_k - S_{k-1}$, and the change time is given by the stopping rule $\min\{k : \sum_{i=0}^{M-1} \mathbb{1}_{\{\delta g_{k-i} \geq h\}} \geq \nu\}$.

Bayesian approaches are available when a priori information is known about the change time, in which case the a priori takes the form of a probability distribution of the change time, see Girshick and Rubin [1952] and Shiryaev [1961].

In the case where the parameters Ω_1 is not available after the change, Wald [1973] proposes two approaches, the first where the likelihood ratio is replaced by a a weighted sum of likelihood ratio over all possible parameters Ω_1, and the second, called generalized likelihood ratio (see Lorden [1971]), where Ω_1 is reimplaced by its maximum likelihood estimate.

4.2.1.2 The CUSUM algorithm

CUSUM

The CUSUM algorithm Page [1954] has been introduced to sequentially detect a change in the mean of a distribution, and also relies on the calculation of the log-likelihood ratio (4.1).

As depicted in proposition 4.1, under the null hypothesis $\Omega = \Omega_0$, the cumulative sum $\sum_{t=0}^{M-1} s_t$ has a negative drift, and under the alternative hypothesis $\Omega = \Omega_1$, it has a positive drift. This property is illustrated on Figure 4.3, where the cumulative sum of s_t is displayed. The data used are the same as for the Figure 4.1. We can see the negative drift for first 50 samples, and the positive drift for the last 50 samples.

Therefore, the relevant information lies in the difference between the current value of cumulative sum of log-likelihood ratios and its minimum value in the past. The decision function is thus defined as

$$S_t = \sigma_t - m_t, \tag{4.2}$$

where $\sigma_t = \sum_{r=0}^{t} s_r$ is the cumulative sum of the log-likelihood ratios and $m_t = \min_{0 \leq r \leq t} \sigma_r$ is the minimum value of the past cumulative sums. A equivalent recursive formulation is to set $S_0 = 0$

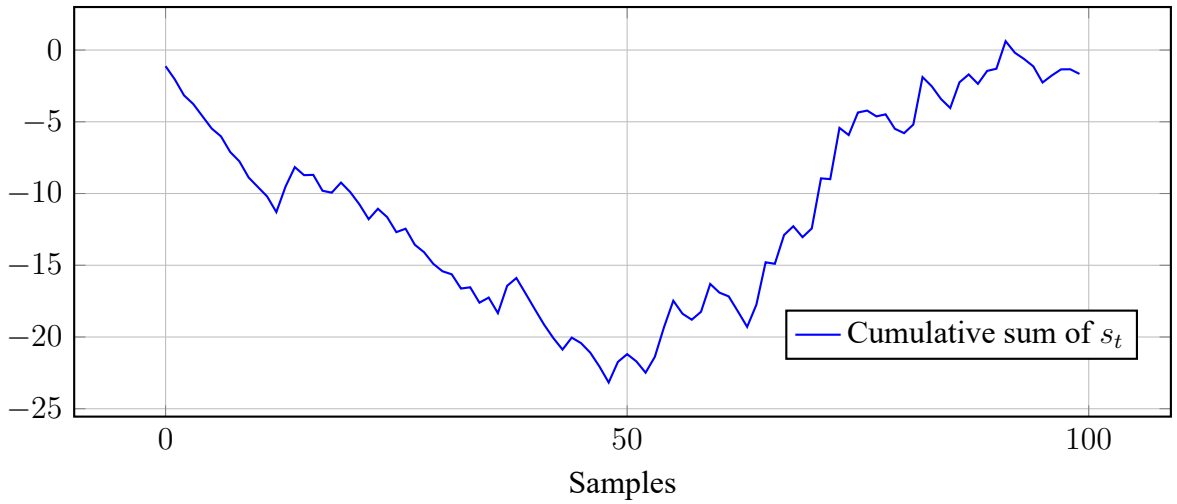


Figure 4.3: Cumulative sum of the log-likelihood ratio $s_t = \log \frac{p_{\Omega_1}(X^{(t)})}{p_{\Omega_0}(X^{(t)})}$ calculated using a hundred Gaussian samples $\{X^{(t)}\}_{t=0...99}$. This samples are the same as the ones used in Figure 4.1: the first fifty are drawn from $\mathcal{N}(0,1)$ and the last fifty are drawn from $\mathcal{N}(2,1)$. We can clearly see the change of drift around $t = 50$ indicating the change time.

and to define $S_t,\ t \geq 1$ recursively by

$$S_t = (S_{t-1} + s_t)^+, \tag{4.3}$$

where z^+ is the positive part of z, that is $z^+ = \max(0, z)$ for any real z. The behaviour of the decision functions (4.2) and (4.3) are displayed in Figure 4.4. The change time is thus defined by thresholding this decision function.

4.2.2 Localising anomalies using the CUSUM algorithm

In this section, we show how we adapt this algorithm to localise anomalies from a sequence of new data $(X_{\mathcal{C}}^{(t)}, X_{\mathcal{Q}}^{(t)}), t = 1, 2, \ldots$, assuming that a reference model Ω_0 has already been learned using normal data, i.e., data that does not contain anomalies.

So far, we assumed that the parameters Ω_0 and Ω_1 of the underlying densities before and after the change time t_0 are known. The chapter 3 provides the materials for estimating a model $\hat{\Omega}_0$ using normal data, i.e., data that does not contain anomalies. We will explain in the following how to choose the parameters Ω_1 after the change time.

Anomaly localisation

Two-sided CUSUM

Remember from Definition 4.1 that localising anomalies means detecting a change in the conditional means of each variables given the others. Since this change might be an increase or a decrease, we use the two-sided CUSUM algorithm as proposed in Basseville et al. [1993], and we will use two different alternative densities, either for detecting an increase or a decrease. Similarly to the CUSUM algorithm, for each t and each variable X_i, $i \in \mathcal{C} \cup \mathcal{Q}$, we define the

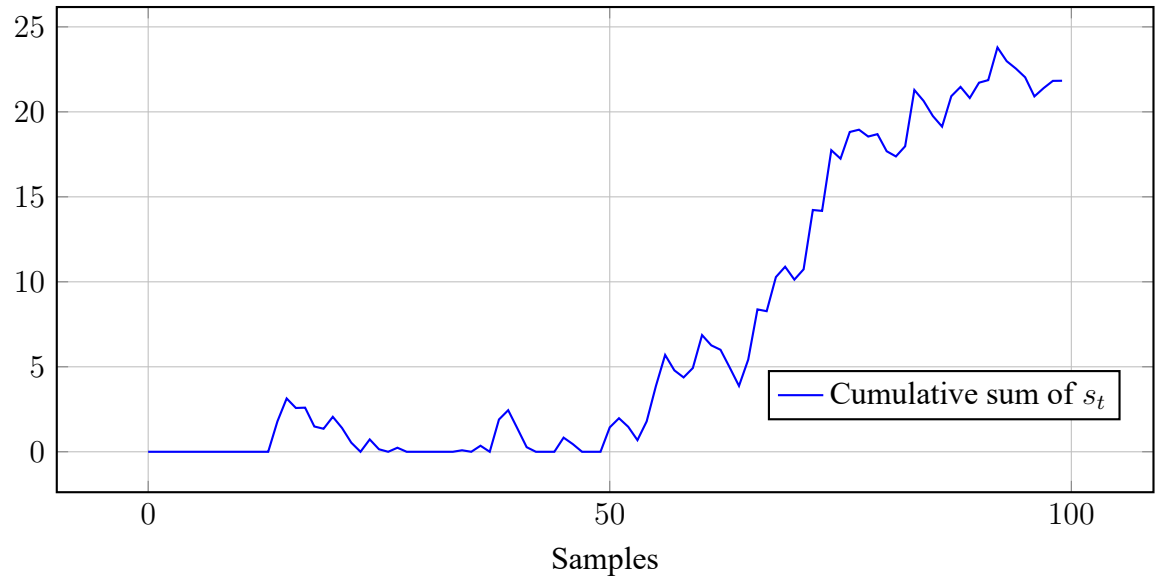


Figure 4.4: Evolution of the decision function defined by (4.3), using the same data as for Figure 4.1 i.e., the first fifty samples are drawn from $\mathcal{N}(0,1)$ and the last fifty from $\mathcal{N}(2,1)$.

instantaneous conditional log-likelihood ratio

$$s_i^{(t)} = \log\left(\frac{p_{\Omega_1}\left(X_i^{(t)}|X_{-i}^{(t)}\right)}{p_{\Omega_0}\left(X_i^{(t)}|X_{-i}^{(t)}\right)}\right), \tag{4.4}$$

where $X_{-i} = \{X_j, j \in \mathcal{C} \cup \mathcal{Q}, j \neq i\}$ and p_{Ω_1} is the density under the alternative hypothesis. We also defined recursively a decision statistic by $S_i^{(0)} = 0$ and

$$S_i^{(t)} = \left(S_i^{(t-1)} + s_i^{(t)}\right)^+, \qquad t = 1, 2, \dots, \tag{4.5}$$

where $(z)^+ = \max(z, 0)$. Here p_0 and p_1 denote respectively the density of the null hypothesis with parameters Ω_0 and the alternative hypothesis with parameters Ω_1, that is, the conditional density of the targeted anomalous behaviour.

The use of the conditional distributions in equation (4.4) actually yields a localisation algorithm. As it was discussed in section 3.1.3, for two arbitrary variables X_i and X_j with $i \neq j$, $i, j \in \mathcal{C} \cup \mathcal{Q}$, the entries associated with node potentials – i.e., $\{\Theta_{ii}\}_{i\in\mathcal{C}}$, $\{\mu_i\}_{i\in\mathcal{Q}}$ and $\{\Delta_{ii}\}_{i\in\mathcal{Q}}$ – only parametrise the conditional distribution of x_i given the remaining variables x_{-i}, and the entries associated to edge potentials – i.e., all the entries of Φ plus the non diagonal entries of Θ and Δ – only parametrise the conditional distributions $p_\Omega(x_i|x_{-i})$ and $p_\Omega(x_j|x_{-j})$. Hence, a change in a parameter ω will only impact the conditional distribution(s) parametrised by ω.

In contrast, the marginal distribution $p_\Omega(x_i)$ of any variable x_i is parametrised by all the entries of Ω. Hence, a modification of any parameter of the model will impact all the marginal

distributions, what prevents the localisation of the changed parameter.

So far, we have assumed that p_{Ω_0} and p_{Ω_1} were known. On the one hand, Ω_0 is learned using normal data available before hand. On the other hand, Ω_1 can not be known in advance and we now need to find a sensible approach to set the density under the alternative. Since we monitor conditional likelihood ratios, we define each conditional density $p_{\Omega_1}(x_i|x_{-i})$ separately, depending on the type of the variable x_i, that is, either quantitative or categorical.

4.2.2.1 The alternative density for quantitative variables

We focus first on the quantitative variables. By the proposition 3.1.i, the conditional distribution of $X_{\mathcal{Q}}^{(t)}$ given $X_{\mathcal{C}}^{(t)}$ is the multivariate Gaussian $\mathcal{N}(\nu^{(t)}, \Delta^{-1})$, with $\nu^{(t)} = \Delta^{-1}(\mu + \Phi^T X_{\mathcal{C}}^{(t)})$. It follows that, for all $i \in \mathcal{Q}$, the conditional distribution of $X_i^{(t)}$ given $X_{-i}^{(t)}$ is Gaussian univariate with mean

$$e_i^{(t)} = \mathbb{E}_\Omega[X_i^{(t)} \mid X_{-i}^{(t)}] = \Delta_{i,-i}^{-1}\Delta_{-i,-i}\left(X_{\mathcal{Q}-i}^{(t)} - \nu_{-i}^{(t)}\right) + \nu_i^{(t)}$$

and variance

$$\sigma_i^2 = \mathrm{Var}_\Omega(X_i^{(t)} \mid X_{-i}^{(t)}) = \Delta_{ii}^{-1} - \Delta_{i,-i}^{-1}\Delta_{-i,-i}\Delta_{-i,i}^{-1}.$$

Observe that $e_i^{(t)}$ depends on the data, whereas it is not the case for σ_i, which only depends on the parameter Δ.

Intuitively, in the case where there is only one quantitative variable, under the null hypothesis, the data have a Gaussian distribution with mean ν and variance σ^2. We define the alternative densities as shifts of $\pm\delta\sigma$ of the conditional density under the null hypothesis, as depicted in Figure 4.5. The parameter δ is controlling the sensitivity of the anomaly detection. On the one hand, the larger δ, the more anomalous the change must be to make the likelihood ratio positively drifted and then eventually trigger a detection. On the other hand, the smaller δ, the less negative the drift is when no change occurs, increasing the probability of a false alarm.

In the general case, for each quantitative variable X_i, $i \in \mathcal{Q}$, we want to detect a change in $p_\Omega(X_i^{(t)} \mid X_{-i}^{(t)})$. We define the conditional density $p(X_i^{(t)} \mid X_{-i}^{(t)})$ of the alternative hypothesis as a Gaussian density with same variance σ_i^2 and a modified mean $e_i^{(t)} \pm \delta\sigma_i$. The log-likelihood ratio (4.4) then becomes, for $i \in \mathcal{Q}$,

$$\begin{aligned} s_i^{(t)\pm} &= \frac{1}{2}\left(\frac{X_i^{(t)} - e_i^{(t)}}{\sigma_i}\right)^2 - \left(\frac{X_i^{(t)} - (e_i^{(t)} \pm \delta\sigma_i)}{\sigma_i}\right)^2 \\ &= \pm\frac{(X_i^{(t)} - e_i^{(t)})}{\sigma_i}\delta - \frac{1}{2}\delta^2. \end{aligned} \tag{4.6}$$

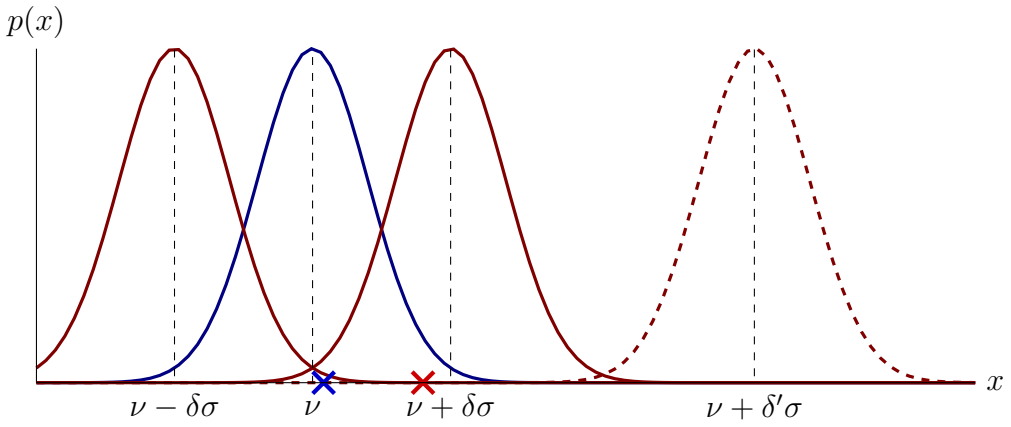


Figure 4.5: Alternative density in dimension 1 for a Gaussian variable. Under H_0, the normal (conditional) density is a univariate Gaussian density (in blue), and under H_1 the two alternative densities (in red) are defined by translating the blue curve by $+\delta\sigma$ or $-\delta\sigma$, corresponding respectively to an increase and a decrease of the mean. The dashed red Gaussian corresponds to an increase of the mean with a different $\delta' > \delta$. The crosses correspond to two samples, for which we want to decide whether they are drawn from the blue Gaussian or not. Intuitively, when the alternative is a Gaussian density with mean $\nu + \delta\sigma$, we can assess that the red dot is more likely drawn from the right red Gaussian than from the blue one, when the blue sample is more likely drawn from the blue Gaussian. On the contrary, when the alternative is defined with a greater mean shift $+\delta'\sigma$ (corresponding to the dashed red Gaussian), then both red and blue are more likely drawn from the blue Gaussian.

In term of $s_i^{(t)+}$ and $s_i^{(t)-}$, this definitions yields two statistics $S_i^{(t)+}$ and $S_i^{(t)-}$ in (4.5), for detecting respectively an increase and a decrease of the conditional mean $e_i^{(t)}$. Later in our experiments, we will consider the sum $\bar{S}_i^{(t)} = S_i^{(t)+} + S_i^{(t)-}$ in order to detect a change in both possible directions.

Note that the parameter δ has an interesting geometric interpretation. By (4.6), the conditional negative drift under the null hypothesis (when no changes occur) of the decision statistic (4.5) is given by $\mathbb{E}_\Omega[s_i^{(t)}|X_{-i}^{(t)}] = -\delta^2/2$. This property is illustrated on the Figure 4.4, where the drift toward zero is controlled by δ.

4.2.2.2 The alternative density for categorical variables

We focus now on the categorical variables. As explained in section 3.1.3, each categorical variable $X_i, i \in \mathcal{C}$ has a conditional Bernoulli distribution with mean

$$p_i = \mathbb{E}_\Omega[X_i \mid X_{-i}] = \frac{\mathrm{e}^{q_\Omega(X,i)}}{1 + \mathrm{e}^{q_\Omega(X,i)}},$$

where

$$q_\Omega(X, i_0) = \theta_{i_0 i_0} + 2\Theta_{i_0,-i_0} X_{-i_0} + \Phi_{i_0,\mathcal{Q}} X_{\mathcal{Q}}.$$

Specifically for the categorical variables, we define the conditional distribution of the alternative hypothesis as a Bernoulli distribution with mean $a_i^{(t)}$. The instantaneous log-likelihood ratio is then given by

$$s_i^{(t)} = X_i^{(t)} \log \frac{a_i^{(t)}}{p_i^{(t)}} + (1 - X_i^{(t)}) \log \left(\frac{1 - a_i^{(t)}}{1 - p_i^{(t)}} \right). \tag{4.7}$$

From there, there are two ways of fixing the parameters $\alpha_i^{(t)}$. First, we choose $a_i^{(t)}$ such as the drift of the decision function (4.5) under the null hypothesis is set to the same value $-\frac{\delta^2}{2}$, as for quantitative variables in (4.6). This drift is given by computing $\mathbb{E}_\Omega[s_i^{(t)} \mid X_{-i}^{(t)}]$ with $s_i^{(t)}$ as in (4.6), yielding the equation

$$p_i^{(t)} \log \frac{a_i^{(t)}}{p_i^{(t)}} + (1 - p_i^{(t)}) \log \left(\frac{1 - a_i^{(t)}}{1 - p_i^{(t)}} \right) = -\frac{\delta^2}{2}. \tag{4.8}$$

It is easy to show that this equation in $a_i^{(t)}$ (with δ and $p_i^{(t)}$ fixed) has two distinct solutions $a_i^{(t)+} \in [p_i^{(t)}, 1]$ (associated to the statistic $S_i^{(t)+}$) and $a_i^{(t)-} \in [0, p_i^{(t)}]$ (associated to $S_i^{(t)-}$), detecting respectively increase and decrease of the mean $p_i^{(t)}$, with a conditional negative drift $-\frac{\delta^2}{2}$ under the null hypothesis, see Figure 4.6. For the same reasons as with quantitative variables, we will consider the sum $\bar{S}_i^{(t)} = S_i^{(t)+} + S_i^{(t)-}$ in the experiments.

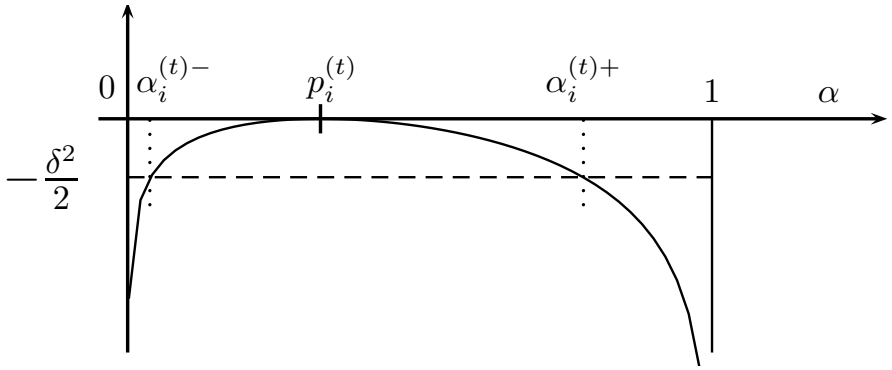


Figure 4.6: Evolution of the negative drift of a categorical variable as a function of $\alpha \in]0, 1[$, when the mean $p_i^{(t)}$ of the conditional Bernoulli distribution of x_i is fixed to $p_i^{(t)} = \frac{1}{3}$. There are two values of α, $\alpha_i^{(t)-} \in]0, p_i^{(t)}[$ and $\alpha_i^{(t)+} \in]p_i^{(t)}, 1[$ yielding a negative drift of $-\frac{\delta^2}{2}$.

Another possible choice is to fix the parameters $\alpha_i^{(t)}$ in order to have a conditional variance $\mathrm{Var}_\Omega \left[s_i^{(t)} | s_{-i}^{(t)} \right]$ equal to one, as it is the case for the quantitative variables in section 4.2.2.1. By (4.7), we have that

$$\mathrm{Var}_\Omega[s_i^{(t)} | s_{-i}^{(t)}] = \mathrm{Var}[X_i^{(t)} | X_{-i}^{(t)}] \left(\log \frac{a_i^{(t)}}{p_i^{(t)}} - \log \frac{1 - a_i^{(t)}}{1 - p_i^{(t)}} \right)^2.$$

Since x_i has a conditional Bernoulli distribution, $\mathrm{Var}[X_i^{(t)}|X_{-i}^{(t)}] = p_i^{(t)}(1-p_i^{(t)})$ and we have that

$$\mathrm{Var}_\Omega[s_i^{(t)}|s_{-i}^{(t)}] = p_i^{(t)}(1-p_i^{(t)}) \left(\log \frac{a_i^{(t)}}{1-a_i^{(t)}} - \log \frac{p_i^{(t)}}{1-p_i^{(t)}} \right)^2 . \tag{4.9}$$

We want $\mathrm{Var}_\Omega[s_i^{(t)}|s_{-i}^{(t)}] = 1$ for any fixed $p_i^{(t)} \in]0,1[$. By (4.9), we have that

$$\frac{a_i^{(t)}}{1-a_i^{(t)}} = \exp\left(\pm \sqrt{\frac{1}{p_i^{(t)}(1-p_i^{(t)})}} \right) \cdot \frac{p_i^{(t)}}{1-p_i^{(t)}},$$

which finally yields two solutions $a_i^{(t)-}$ and $a_i^{(t)+}$, depending on the sign of the term in the exponential, given by

$$\begin{aligned} a_i^{(t)-} &= \left[1 + \frac{1-p_i^{(t)}}{p_i^{(t)}} \exp\left(+ \frac{1}{\sqrt{p_i^{(t)}(1-p_i^{(t)})}} \right) \right]^{-1}, \\ a_i^{(t)+} &= \left[1 + \frac{1-p_i^{(t)}}{p_i^{(t)}} \exp\left(- \frac{1}{\sqrt{p_i^{(t)}(1-p_i^{(t)})}} \right) \right]^{-1}. \end{aligned} \tag{4.10}$$

The Figure 4.7 shows the evolution of the drift seen as a function of $p_i^{(t)} \in]0,1[$.

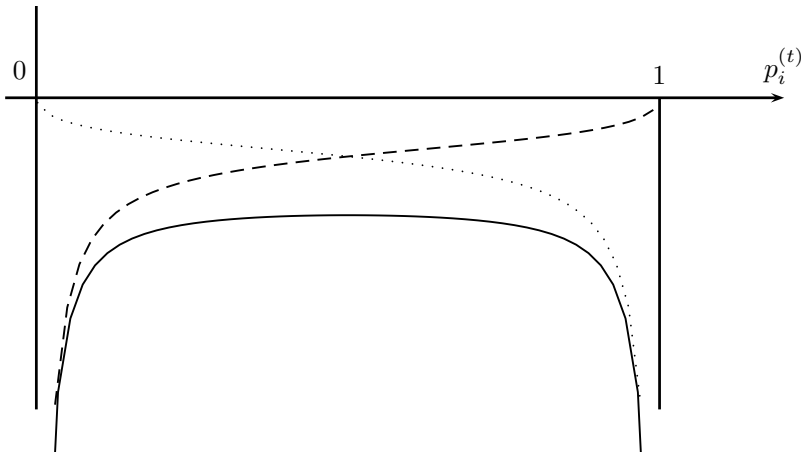


Figure 4.7: Evolution of the negative drift for a categorical variable x_i when the parameters $a_i^{(t)+}$ and $a_i^{(t)-}$ are given by (4.10), so that the conditional variance $\mathrm{Var}_\Omega[S_i^{(t)}|s_{-i}^{(t)}]$ is equal to one. The drift for $S_i^{(t)-}$ is the dotted line, the drift for $S_i^{(t)+}$ is the dashed line and the drift for $\bar{S}_i^{(t)}$ is the continuous line.

Algorithm 8 Two-sided CUSUM for anomaly detection and localisation

Input A learned model p_Ω, a sensibility parameter δ, a threshold h, and a dataset or a stream $\{X^{(t)}\}_t$,

Initialize $S_i^{(0)} = 0$ for each $i \in \mathcal{C} \cup \mathcal{Q}$.
for each sample $X^{(t)}$, **do**
  for each $i \in \mathcal{Q}$, **do**
    Update $S_i^{t+1} = (S_i^{(t)} + s_i^{(t)})^+$ using the equation (4.6).
  for each $i \in \mathcal{C}$, **do**
    Compute the mean $a_i^{(t)}$ of the alternative Bernoulli distribution using equation (4.8),
    Update $S_i^{t+1} = (S_i^{(t)} + s_i^{(t)})^+$ using the equation (4.7).
For each variable x_i, find the change time $\tau_i = \min\{t; S_i^{(t)} > h\}$.

Return the change times $\{\tau_i, i \in \mathcal{C} \cup \mathcal{Q}\}$.

4.2.2.3 Our two-sided CUSUM algorithm and calibration of its parameters

Our two-sided CUSUM is given in Algorithm 8 for detecting and localising anomalies in data. Used with a batch file of M samples, this algorithm has a time complexity $\mathcal{O}(M \cdot |\mathcal{C}| \cdot |\mathcal{Q}|)$, in the case where we use the parameters $a_i^{(t)}$ defined by (4.10). Note that in the case where we would like to fix the drift instead of the conditional variance, there is no closed-form of the inverse of the drift (4.8) and thus it has to be approximated using for instance a dichotomy method or by preloading a table associating, with δ fixed, the parameter a of the anomalous density given the parameter p of the density under H_0.

The Algorithm 8 has two tuning parameters that the user has to set:

- the sensitivity parameter δ,
- the detection threshold h.

Under the null hypothesis, each decision statistic $S_i^{(t)+}$ or $S_i^{(t)-}$ evolves with a negative drift $-\delta^2/2$. Hence, because of the positive part in (4.5), it remains close to zero with high probability. In contrast, under the alternative, the conditional drift becomes positive and the decision statistic $\bar{S}_i^{(t)}$ eventually increase above any arbitrarily high threshold h. We thus label as a change time the first times t when $\bar{S}_i^{(t)} > h$. The choice of δ sets how sensitive the test is to a close alternative, while the choice of h is a compromise between the false alarm probability over a given horizon and the delay needed to raise an alarm after a change of distribution. Finally and most interestingly, the set of indices i for which the alarm is raised provides a way to identify the variables for which not only the marginal distribution has changed but also the conditional one, given all other available variables.

Fixing the detection threshold is a complex task, and the most classic ways require the computation of the the *detection delay in the worst case* (Lorden [1971]) defined by

Detection delay

$$\bar{\mathbb{E}}_{\Omega_1}[t_0] = \sup_{t \geq 0} \left\{ \text{ess sup } \mathbb{E}_{\Omega_1} \left[(t_0 - t + 1)^+ \mid X^{(0)}, \ldots, X^{(t)} \right] \right\},$$

where ess sup denotes the essential supremum, Ω_1 is the parameter of the model under the alternative hypothesis and t_0 is the change time (see section 4.2.1), or the *average run length* function (ARL, Page [1954]), defined as

Average Run Length

$$\text{ARL} = \mathbb{E}_{\Omega}[t_0],$$

where t_0 is the change time. Depending on the value taken by Ω, the ARL function can take two particular interesting values.

- If $\Omega = \Omega_0$, then the average run length under the null hypothesis becomes $\text{ARL}_0 = \mathbb{E}_{\Omega_0}[t_0]$, i.e., the expected number of samples before a false alarm is raised, and can be also viewed as the average time between two false detections. We thus want this quantity to be as large as possible to minimise the false detection rate.
- If $\Omega = \Omega_1$, then $\text{ARL}_1 = \mathbb{E}_{\Omega_1}[t_0]$ is an expected number of samples before detecting a change and can also be viewed as a average detection delay. We want this quantity to be as small as possible to minimise the reaction time of the algorithm.

We refer to Lorden [1971], Moustakides [1986], Ritov [1990] and Basseville et al. [1993] for precise results on the computation of these delays under simple assumptions, when the CUSUM algorithm is computed with the standard conditional likelihood.

4.3 Application to synthetic data

In this section, we present results of anomaly detection and localisation with synthetic data. We suppose here that we have already learned the model parameters Ω from normal data. The data is composed of 50 normal observations sampled from the model using the Algorithm 2, and 50 anomalous observations sampled from an altered model where one parameter value in Ω has been modified.

We use the same model structure as in section 3.3.1, with 4 categorical and 4 quantitative variables. The model is represented in Fig. 4.8, with a colormap that will be kept for all the experiments. The parameters $\Omega^* = (\Theta^*, \mu^*, \Delta^*, \Phi^*)$ of the model are chosen this way:

- Θ^* is a 4×4 square matrix, with -0.5 for the diagonal entries and 0.5 for the entries of the upper and lower diagonals,

- μ^* is a vector of 4 null entries,
- Δ^* is a 4×4 square matrix with 1 for the diagonal entries and 0.25 for the entries of the upper and lower diagonals,
- Φ^* is a 4×4 square matrix with 0.5 for the entries of the diagonal and 0 elsewhere.

$$\Theta^* = \begin{bmatrix} -0.5 & 0.5 & 0 & 0 \\ 0.5 & -0.5 & 0.5 & 0 \\ 0 & 0.5 & -0.5 & 0.5 \\ 0 & 0 & 0.5 & -0.5 \end{bmatrix}, \qquad \mu^* = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\Delta^* = \begin{bmatrix} 1 & 0.25 & 0 & 0 \\ 0.25 & 1 & 0.25 & 0 \\ 0 & 0.25 & 1 & 0.25 \\ 0 & 0 & 0.25 & 1 \end{bmatrix}, \qquad \Phi^* = \begin{bmatrix} .5 & 0 & 0 & 0 \\ 0 & .5 & 0 & 0 \\ 0 & 0 & .5 & 0 \\ 0 & 0 & 0 & .5 \end{bmatrix}.$$

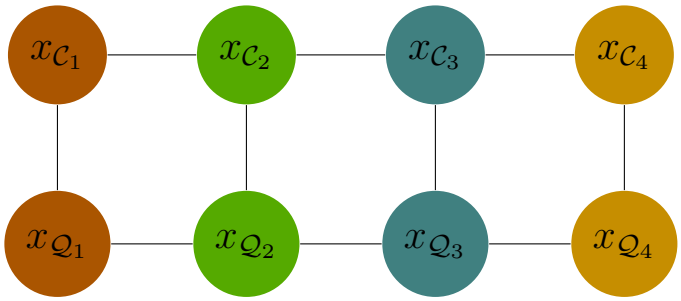


Figure 4.8: Structure of the network used for the experiments. This network uses four binary variables $x_{\mathcal{C}_1}, \ldots, x_{\mathcal{C}_4}$ (on the top row) and four quantitative variables $x_{\mathcal{Q}_1}, \ldots, x_{\mathcal{Q}_4}$ (in grey, on the bottom row).

We have tested three different modifications on the parameters of Ω^* :

1. the conditional distribution of the second (green) quantitative variable is changed by moving μ_2 from 0 to 3,
2. the conditional distribution of the first (brown) categorical variable is changed by moving $\theta_{1,1}$ from -0.5 to -4,
3. the conditional distributions of the first (brown) categorical and third (grey) quantitative variable are changed by moving $\phi_{1,3}$ from 0.5 to 2. Hence in this case the Markov field structure is modified, see Figure 4.9.

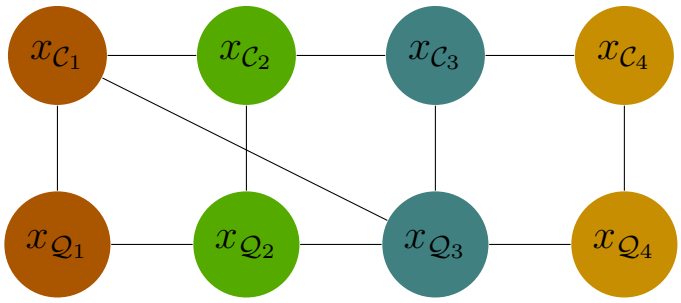


Figure 4.9: Structure of the network used for the third experiment, where an edge was added between the first categorical variable $x_{\mathcal{C}_1}$ and the third quantitative variable $x_{\mathcal{Q}_3}$.

Figure 4.10 shows the temporal evolution of the statistic $\bar{S}_i^{(t)}$ computed for every variable and for the three kinds of anomalies, when we use (4.10) for the parameters under H_1 of the conditional distribution of the categorical variables, corresponding to a fixed variance of the decision function. As expected, the plots on the top row show that when changing μ_1, only the statistic of the green quantitative variable is increasing, indicating that the green variable is carrying alone the change of conditional distribution. The same thing can be concluded for the two others modifications on $\theta_{1,1}$ and $\phi_{1,3}$. These results show that our method correctly detects and localises the changes in the conditional distributions.

Wilcoxon test

We compare our method to the *Wilcoxon test* presented in Lung-Yut-Fong et al. [2011], which is designed to detect changes in the distribution of a set of quantitative variables from a batch dataset $\{X^{(m)}\}$, $m = 1 \dots M$ of M samples. This test is based on the rank statistics computed, for each quantitative variable X_i, $i \in \mathcal{Q}$, as

$$V_i(M_1) = \frac{1}{M^{3/2}} \sum_{r=1}^{M_1} \sum_{s=M_1+1}^{M} \left(\mathbb{1}_{\{X_i^{(r)} \leq X_i^{(s)}\}} - \mathbb{1}_{\{X_i^{(s)} \leq X_i^{(r)}\}} \right), \quad 1 \leq M_1 < M. \tag{4.11}$$

In the following, we apply this approach to detect a change of distribution for each quantitative variable. Figure 4.11 displays the statistic of this test as a function of the possible change times. When only one change occurs in the data, this statistic is expected to approximately have a triangle shape with a maxima or a minima around the true change time. We use the same dataset as for the experiment with the anomalies localised on the second (green) quantitative variable, where μ_1 changes from 0 to 3 at time $t = 50$. Figure 4.11 should thus be compared with the top row of Figure 4.10. The Wilcoxon test is a batch method as it requires the whole set of data to be computed. We observe that, in contrast to our approach, the Wilcoxon test applied individually to each variable is not suited to localise the anomaly since a change of μ_1, although it only modifies the conditional distribution of X_1 given X_{-1}, yields a change of all the marginal distributions. This is why in Figure 4.11, the Wilcoxon statistics displays triangle shapes for all the quantitative variables with a more obvious change for the variables directly connected to X_1.

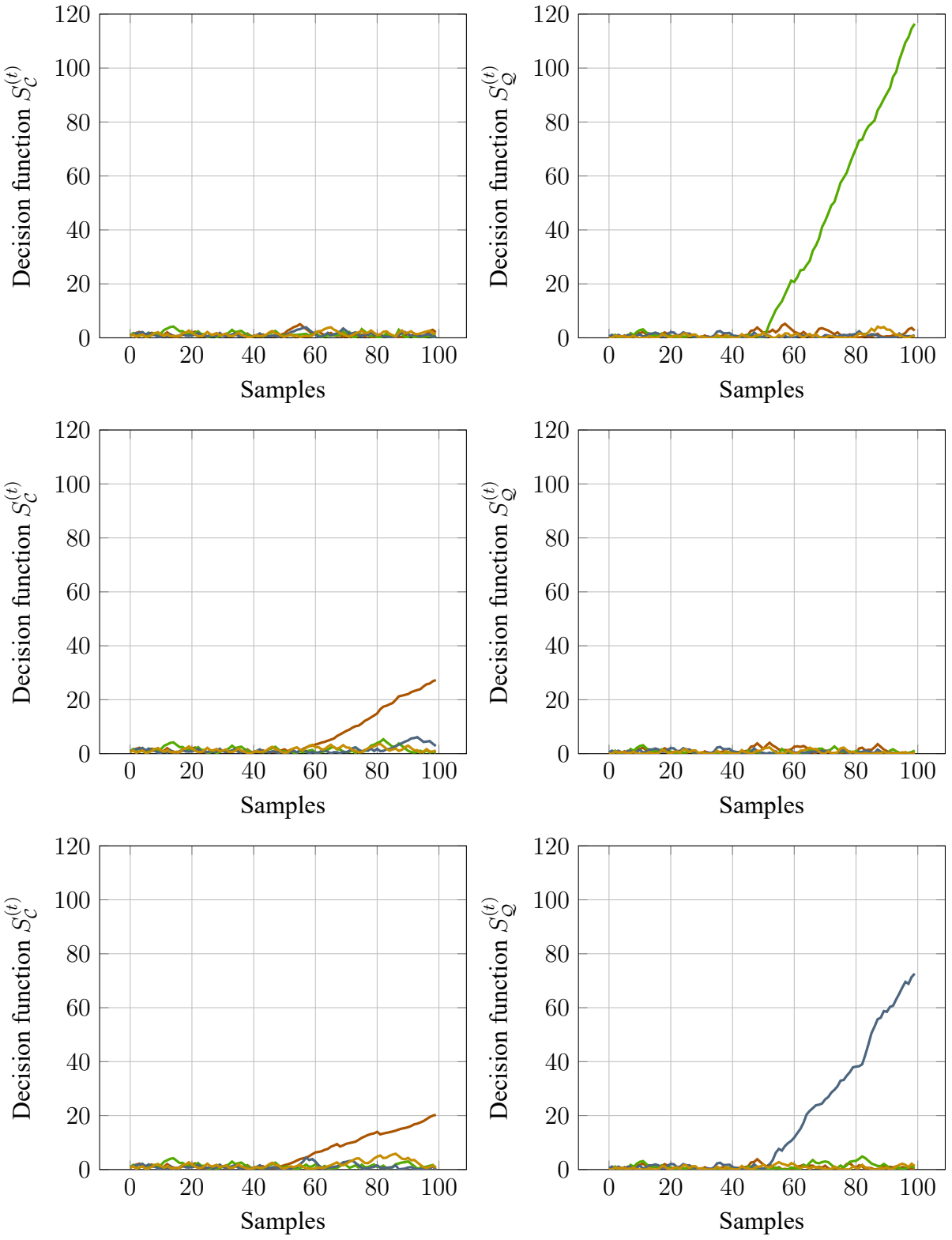


Figure 4.10: Time evolution of $\bar{S}_i^{(t)}$ for quantitative variables on the left and categorical variables on the right, where we have used the parameters (4.10) yielding $\mathrm{Var}_\Omega[s_i^{(t)}|s_{-i}^{(t)}] = 1$ for all $i \in \mathcal{C}$. The colors of the plots correspond to the colors of the variables in the graph of Figure 4.8. Top row : change on μ_2. Middle row : change on $\theta_{1,1}$. Bottom row : change on $\phi_{1,3}$. For each experiment, the first 50 samples are sampled with parameter Ω, and the last 50 samples are sampled with the modified parameter.

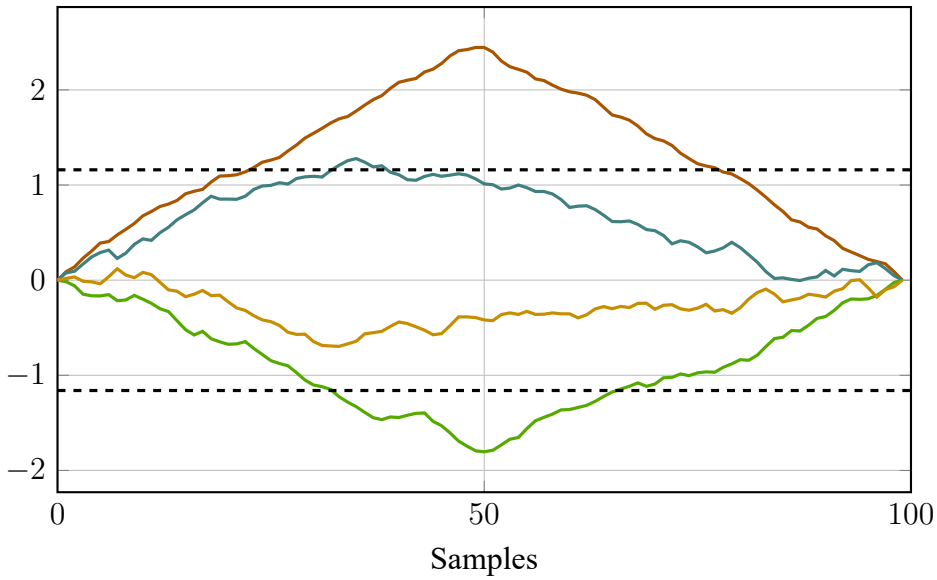


Figure 4.11: Evolution of the Wilcoxon statistic for 100 samples of 4 quantitative variables, with the same color code as for Figure 4.8. After the 50^{th} sample, we have modified Ω with $\mu_1 = 3$. The dashed lines indicate the thresholds for detecting a change with a 5% false detection probability.

We end this experimental section by showing some ROC curves in Figure 4.12. We use the same data as for Figure 4.10, i.e., three datasets of 100 samples, the first fifties being i.i.d. drawn from p_{Ω^*}, and the last fifties are drawn from the same distribution, where a parameter has been changed. For these curves, the false alarms are the points for which, under H_0, the statistics of any variable not impacted by the parameter change is higher than a threshold, and the true positive are the point for which, under H_1, the statistic of the variable impacted by the parameter change is higher than the threshold. The probability of true detection PTD and the probability of false alarm PFA are thus defined, for our CUSUM approach, by

$$\begin{aligned} PTD &= \mathbb{P}(\forall i \in \mathcal{A}, \quad S_i^{(t)} \geq h \mid H_1), \\ PFA &= \mathbb{P}(\exists i \in \mathcal{C} \cup \mathcal{Q} \setminus \mathcal{A} \quad \text{such as } S_i^{(t)} \geq h \mid H_0), \end{aligned}$$

and, for the Wilcoxon statistic, by

$$\begin{aligned} PTD &= \mathbb{P}(\forall i \in \mathcal{A}, \quad |V_i^{(t)}| \geq h \mid H_1), \\ PFA &= \mathbb{P}(\exists i \in \mathcal{C} \cup \mathcal{Q} \setminus \mathcal{A} \quad \text{such as } |V_i^{(t)}| \geq h \mid H_0), \end{aligned}$$

where h denotes a threshold, $\mathcal{A}$ denotes the variables impacted by the parameter change, and $\mathcal{C} \cup \mathcal{Q} \setminus \mathcal{A}$ is the complementary set of $\mathcal{A}$ in $\mathcal{C} \cup \mathcal{Q}$, i.e., the variables not impacted by the change. Namely, for the three changes on μ_2, θ_{11} and ϕ_{13}, the set $\mathcal{A}$ respectively corresponds to $\{x_{\mathcal{Q},2}\}$, $\{x_{\mathcal{C},1}\}$ and $\{x_{\mathcal{C},1}, x_{\mathcal{Q},3}\}$.

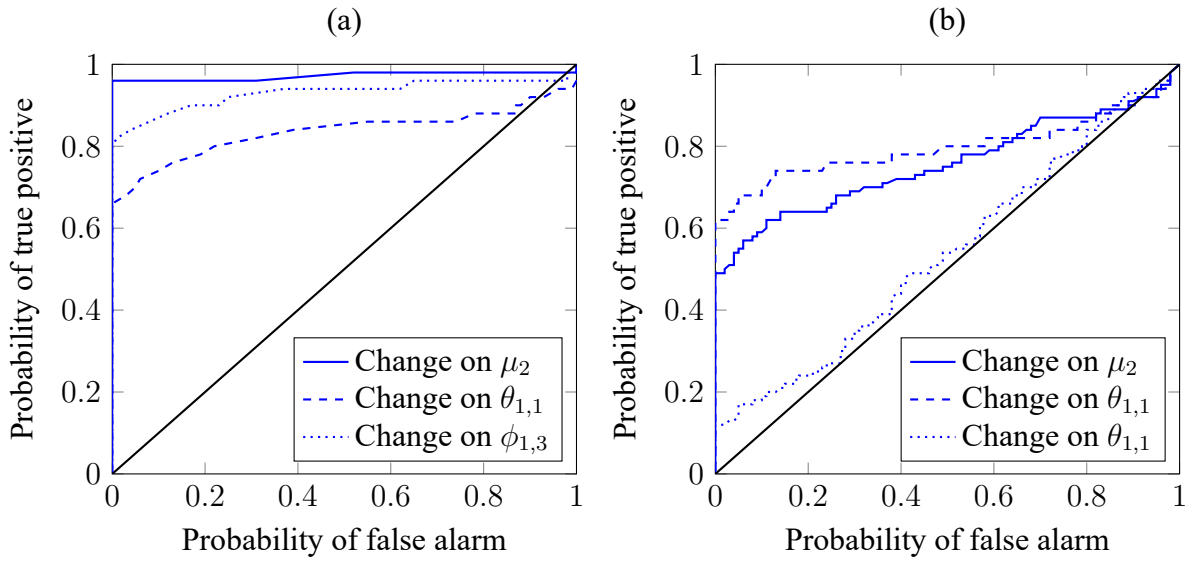


Figure 4.12: ROC curves computed over a set of 100 samples, where the 50 first were drawn from the model Ω^* and the last 50 from Ω^* change on one parameter. The black line indicate the bisector $PTD = PFA$. The graph (a) is obtained using the decision statistics we defined in (4.3), and the graph (b) is obtained using the Wilcoxon statistics (4.11).

Chapter 5

Applications

This chapter presents the experiments we performed for mixed model learning and anomaly localisation using real data provided by Thales. We inform the reader that most of the used names, shown values and times have been slightly altered or totally changed for confidentiality reasons, in a way so that the problems we faced and the choices we made are still completely valid.

5.1 Presentation of the data

The data are produced by the built-in test during the execution of the Touch and go scenario, simulating a series of takeoffs and landings (see Example 2.3) and the all modes scenario (see Example 2.4). Both scenarios are deterministic: a test bench reproduces flight conditions by simulating some predetermined environmental conditions and orders of the pilot. The behaviour of the radar is hence very different depending on the scenario. In particular, a different set of fields and variables is used for each scenario.

Scenario Touch and go This scenario is the smallest in term of involved fields and variables. After applying a dimensionality reduction strategy (as explained in section 2.3.1), we kept 9 frames, 28 fields, and 86 variables. Among the fields, 9 are quantitative and 19 are categorical. These 19 categorical fields correspond to 77 binary variables, i.e., $|\mathcal{C}| = 77$ and $|\mathcal{Q}| = 9$. Following the reassembling process explained in section 2.3.2, each datafile produced during a Touch and go scenario end up with around 1.5 million samples.

Scenario All modes This scenario is the largest scenario in term of involved variables. After reducing the dimension, we ended up with 29 frames, 219 fields and 1004 variables. In particular, among the 219 fields, 49 are quantitative and 170 are categorical. These 170 categorical variables correspond to 955 binary variables, i.e., $|\mathcal{C}| = 955$ and $|\mathcal{Q}| = 49$. Each datafile has around 2 million samples.

Also note that, as discussed in section 2.1.2.3, even if the stimuli are deterministic, the behaviour of the radar is partially random. Many variables may take different values during the tests, and some events (breakdowns, calibrations, …) will change the data produced by the radar. Let's illustrate these facts with some examples.

- The temperature of some pieces of the radar has a very particular behaviour. At start, these temperatures are equal to the temperature of the test room, which may vary depending on the use of the room (particularly if other radars are being tested), the weather or the season of the year. Similarly, the asymptotic temperature (the temperature reached by the radar after a long moment of use) is supposed to be quite constant, but this constant also depends on many environmental parameters and is not the same for every test, even on the same radar. The Figure 5.1 depicts these observations.

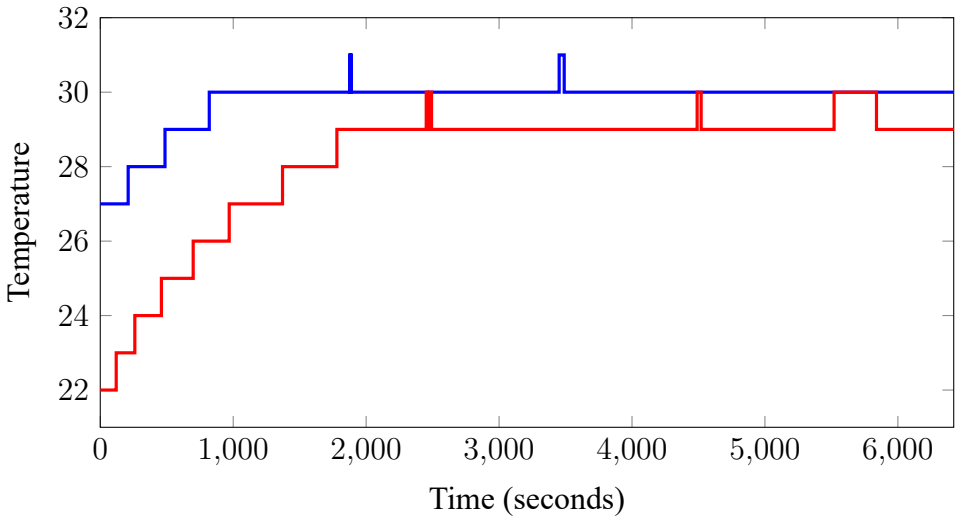


Figure 5.1: Evolution of the temperature of the same component of two different radars, during the execution of two all modes scenarios several months apart. The values have been modified for confidentiality reasons.

  This dependency in the environmental conditions also impacts other quantitative variables. A noteworthy common characteristic of these variables is that they often have a different behaviour during different tests, even for the same radar. The initial temperature depends a lot on environmental conditions, and the asymptotic values are also not guaranteed to be equal. These particularities result in the rise of many false alarms. To address this issue, in the training dataset, we have added a centred noise $\mathcal{N}(0, \sigma_i)$ on each concerned variable x_i. This modification has made the number of false alarm dropped to zero in the cases we studied, while still detecting the true anomalies.

- Remember that, as explained in section 2.3.2, the data have been recovered from the radar with a specific format, which mainly consists of a series of frames containing the values of

a few variables, in the order at which they have been read on the internal communication bus by the recording device. The processing we defined in section 2.3 produces an array where each column corresponds to a variable, and each row to a read frame. The missing data in each row have been mainly filled by copying the last known values. However, the order at which the data have been read necessarily changes between two tests (due to computational load, stacking of queries, …), which results in changes in the data produced by our processing. This observation mainly impacts frames sent at a high frequency, like calibration reports.

When learning a graphical model from data, we will assume that this data does not contain any anomalies. Neither the built-in test nor basic manual investigations have reported any breakdown or anomaly. Our belief in this assumption will be reinforced retrospectively by applying the detection and localisation task on the training data themselves. However, we still can not ensure that the training data doest not contain any anomalies.

For the anomaly detection and localisation part, we will analyse data that may contain breakdowns, i.e., data for which the built-in test has produced breakdown reports. One objective is to retrieve, at least, those reported breakdowns. We will show the results on several test sets produced by the All modes and the Touch and Go scenarios. We will analyse one test set from produced by the Touch and Go scenario, which contains one known breakdown (reported by the built-in test), consisting in a reset of the radar, followed by a wrong activated mode of working. We will also test several test sets produced by the All Modes scenario, for some of which no breakdown has been reported by the built-in test and appears to be well-working.

5.2 Model learning

We will apply the learning techniques we developed in Chapter 3 on real data produced by the RBE2 during the execution of deterministic scenarios. For the scenario Touch and go, Figure 5.2 and Figure 5.3 show respectively the minimisation of the negative log-pseudolikelihood (optimised with Algorithm 3) and the minimisation of the negative log-likelihood (using Algorithm 6). The training set contains around $M = 1.5 \cdot 10^6$ samples of 86 variables, with 9 quantitative variables and 77 binary variables, i.e., $|\mathcal{Q}| = 9$ and $|\mathcal{C}| = 77$, without missing data. Similarly, Figure 5.4 shows the minimisation of the negative the log-likelihood using Algorithm 6, for a training set produced by the All Modes scenario. This training set contains around $M = 2.1 \cdot 10^6$ samples of 1004 variables, among which 49 are quantitative and 955 are binary, i.e., $|\mathcal{Q}| = 49$ and $|\mathcal{C}| = 955$, without missing data.

When using the deterministic proximal gradient algorithm, we used the line search Algorithm 4 to determine the gradient step size γ_t at each iteration. We used $\gamma_0 = 0.1$ and $\beta = 0.5$.

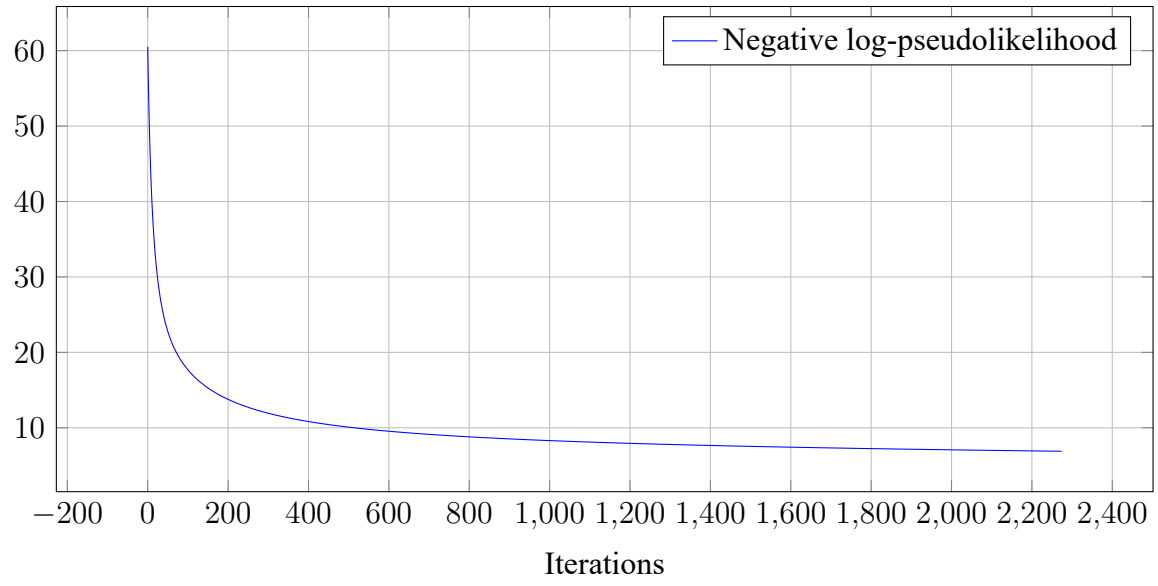


Figure 5.2: Minimisation of the negative log-pseudolikelihood (3.25) during the learning of a mixed model, using the proximal gradient Algorithm 3. The data are produced during the execution of the Touch and go scenario, with 1.7 millions samples of 314 variables.

We have used the following regularisation parameters, as discussed in section 3.3.2:

$$\lambda_\Theta = 1.4\sqrt{\frac{\log(|\mathcal{C}| + |\mathcal{Q}|)}{M}} \qquad \lambda_\Delta = 0.95\sqrt{\frac{\log(|\mathcal{C}| + |\mathcal{Q}|)}{M}} \qquad \lambda_\Phi = 4.6\sqrt{\frac{\log(|\mathcal{C}| + |\mathcal{Q}|)}{M}}.$$

The learning is stopped when the difference of the log-pseudolikelihood between two successive iterations is smaller than a given threshold $\epsilon > 0$, i.e., when

$$|\ell(\Omega_t : \mathcal{D}) - \ell(\Omega_{t+1} : \mathcal{D})| < \epsilon,$$

with $\epsilon = 0.001$ for our experiments.

For the stochastic proximal gradient algorithm, it is complex to define a stopping criterion based on the score change, because of the variability of the estimation of the partition function Z_Ω. We have manually stopped the learning after around one day of learning. Following the discussion made in section 3.2.4, we used a constant gradient step $\gamma_t = 1e^{-4}$ and a linearly increasing Markov chain length $\nu_t = |\mathcal{C}| + |\mathcal{Q}| + t$, for $t > 0$. We used the following regularisation parameters, as discussed in section 3.3.2:

$$\lambda_\Theta = 3.7\sqrt{\frac{\log(|\mathcal{C}| + |\mathcal{Q}|)}{M}} \qquad \lambda_\Delta = 3.2\sqrt{\frac{\log(|\mathcal{C}| + |\mathcal{Q}|)}{M}} \qquad \lambda_\Phi = 4.3\sqrt{\frac{\log(|\mathcal{C}| + |\mathcal{Q}|)}{M}}.$$

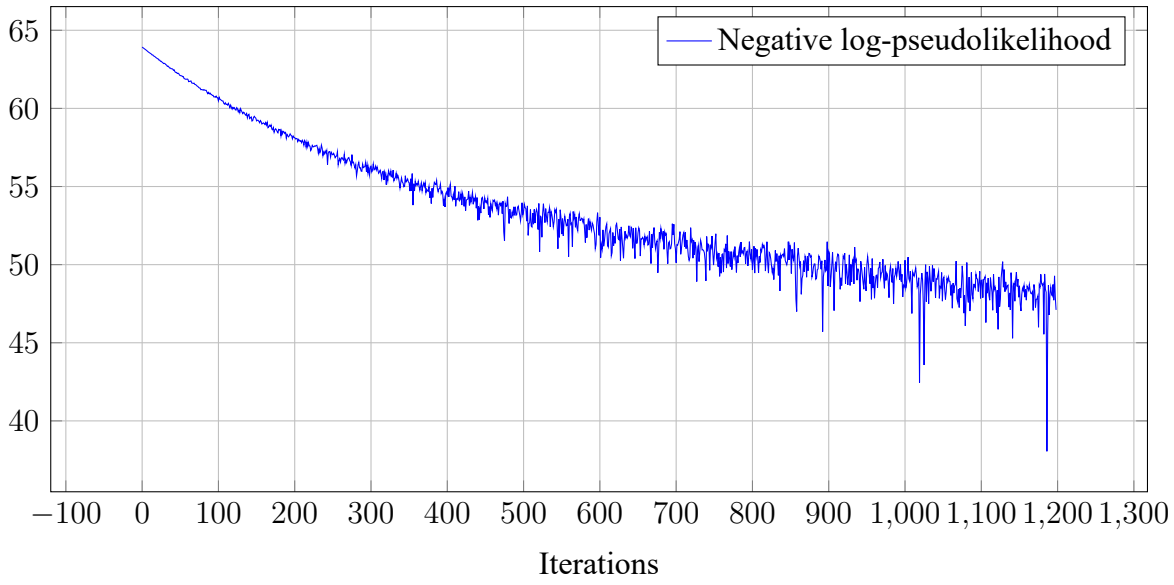


Figure 5.3: Minimisation of the negative log-likelihood (3.19) during the learning of a mixed model, using the stochastic proximal gradient Algorithm 6. The data are produced during the execution of the Touch and go scenario, with 1.7 millions samples of 314 variables.

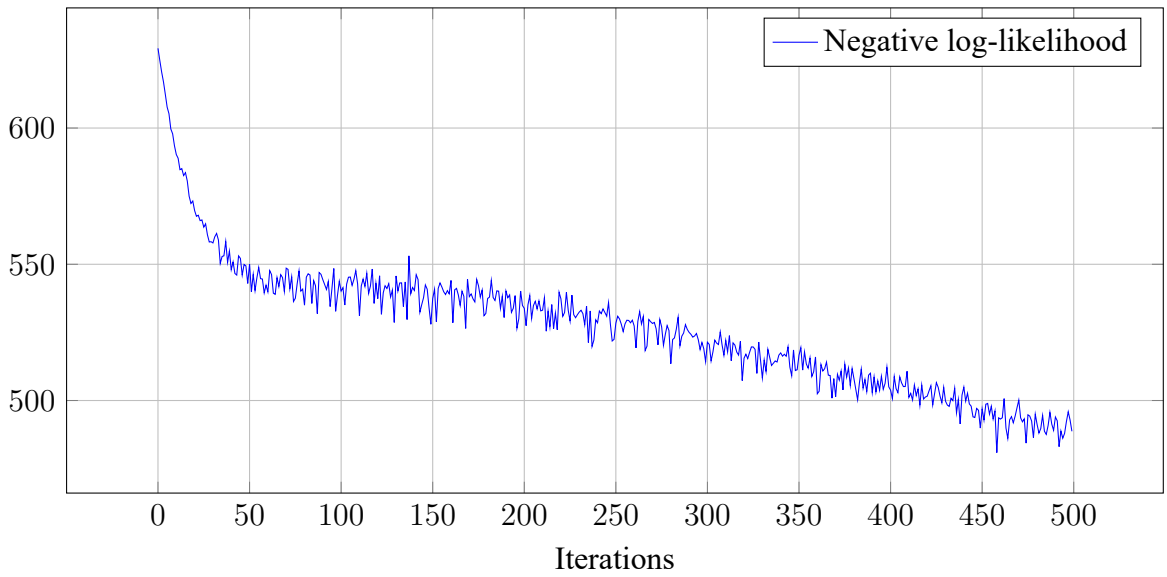


Figure 5.4: Minimisation of the negative log-likelihood (3.19) during the learning of a mixed model, using the stochastic proximal gradient Algorithm 6. The data are produced during the execution of the scenario All modes, with 2 millions samples of 1004 variables.

5.3 Anomaly detection and localisation

We present the results of the anomaly detection and localisation of new records of the scenario Touch and go and All modes, using the models learned with good data. For the scenario Touch and go, we are using the model learned by optimising the regularised negative log-pseudolikelihood with deterministic proximal gradient, and for the scenario All modes, we are using the model learned by optimising the penalised negative log-likelihood using stochastic proximal gradient.

5.3.1 Setting the detection thresholds

As highlighted in section 4.2.2.3, the definition of the detection threshold is a complex task. In this work, we decided to fix these thresholds variable by variable to the maximum values reached by the decision statistics on the training set. This decision is motivated by several empirical observations.

- First, the reported anomalies have to contain as little false alarm as possible: it is not worth to have all the true breakdowns detected, if alongside too much false alarms are also detected. Each anomaly will trigger an investigation by an expert, and this investigation time has to be spent mainly on major anomalies or breakdowns. We thus rejected threshold strategies that are based on false alarm probability.
- Even if the log-likelihood ratios (4.6) and (4.7) (with parameters (4.10) for the categorical variables) have been designed to have a conditional variance equal to one, and even under the assumption that the training data are all normal, the conditional probabilities of some samples are sometimes still low, and thus may be labelled as anomalies. This is particularly true for the All Mode scenario, where such an anomaly is visible at each switch of mode, as depicted in Figure 5.5.

Note that fixing the thresholds to the maximum values strongly relies on the assumption that the training data does not contain anomalies, otherwise thresholds might be fixed at a too high value, and true anomalies in new datasets will remain undetected.

5.3.2 Anomaly localisation

5.3.2.1 Analysis of the data produced by the Touch and Go scenario

The radar tested using a Touch and Go scenario came up with some breakdowns, some of which have been reported by the built-in test. According to this report, they consist in a reset of the radar after around one hour of run, followed by a wrong radar mode activation. This wrong mode will

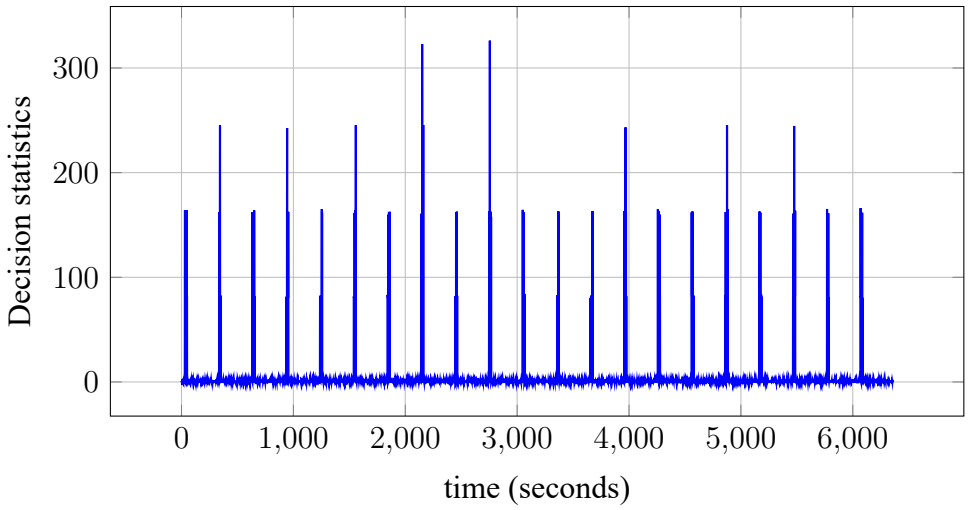


Figure 5.5: The decision statistic of some variable will behaves anomalously when switching modes, but these events are normal from a radar point of view and should not be considered as anomalies.

change the behaviour of many (but not all) variables: most of the quantitative variables are not impacted, in contrast to many of the categorical variables.

The anomaly is detected and localised by thresholding the decision function (4.5), where the thresholds have been fixed variable by variable to the maximum value of their respective decision statistics on the training set. For the training data and the test data, we used the value of the sensitivity parameter $\delta = 1$ for all variables. The Figure 5.6 shows the decision function (4.5) for an categorical variable renamed A (the real name being confidential) during 25 seconds around the detected anomaly, which corresponds to the reset of the radar. We see the score largely exceeding the threshold around the 887 000th sample. In addition, many smaller anomalies are detected after the reset, corresponding to the wrong configuration of the radar. The Figure 5.6 also shows the decision statistic of another quantitative variable renamed B, corresponding to a sensor measuring a temperature on the antenna. In contrast to the variable A, no anomaly is detected, since the decision function $S_B^{(t)}$ remains below the threshold for all samples in the dataset.

The analysis of the decision statistics has enabled the localisation of the main cause of the reset of the radar. However, the activation of the wrong mode has made a lot of variables been anomalous during the rest of test, what has hidden the switch to the wrong mode, in addition to potential new anomalies.

5.3.2.2 Analysis of the data produced by the All Modes Scenario

We tested several records from the All Modes scenario. Again, the thresholds values have been fixed to the maximal values of the decision statistics on the training set. For the training set and the test sets, we fixed the sensitivity parameter to $\delta = 1$ for all variables.

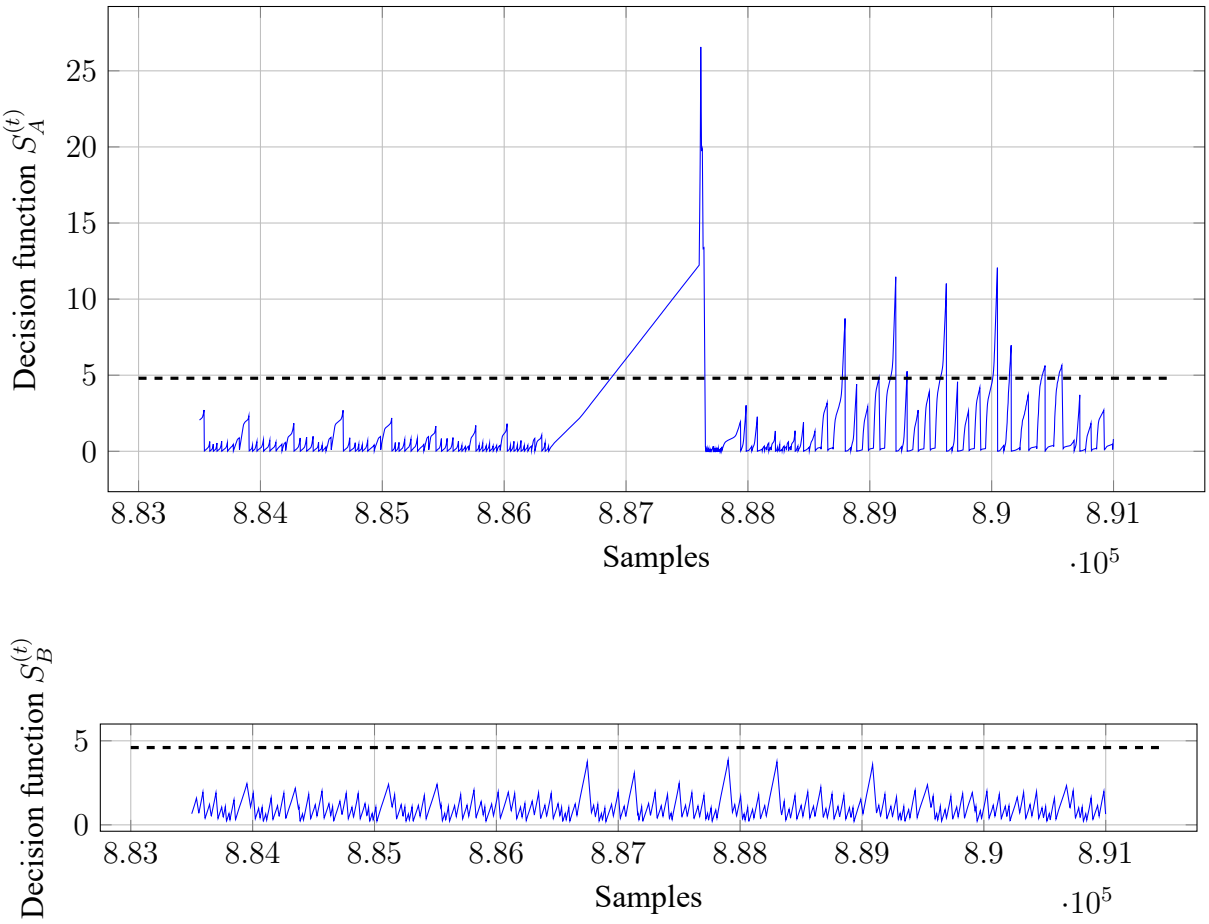


Figure 5.6: Decision functions (4.5) $S_A^{(t)}$ and $S_B^{(t)}$ of an arbitrary categorical variable A and an arbitrary quantitative variable B among the variables used by the scenario Touch and go. They are computed using new records of the scenario Touch and go, where only the samples with index between 883 500 and 891 000 are displayed. This window corresponds to a time window of 25 seconds, centred around a detected anomaly.

Some of the breakdowns occurring during these tests have been detected by the built-in test. In these cases, because of the filtering system (see section 2.1.1), the anomalies are localised on low-level variables alongside on high-level variables. Figure 5.7 shows the decision statistics for a case where an anomaly has been localised on two categorical variables : the first one is a bit of working and is the root cause of the anomaly, and the second one is a signature, i.e., a categorical information summing up other low-level variables.

For all the cases we studied, when a breakdown report were produced by the built-in test, we localised anomalies on the high-level reported variables, alongside on low-level variables that were the root causes of these anomalies. Similarly, we get a low rate of false alarms among the anomalies undetected by the built-in test. This false alarms mainly come from the short test starting period, when all the variables are still warming up and haven't taken their normal values. This is typically the case for the temperature depicted in Figure 5.1, and give rise to

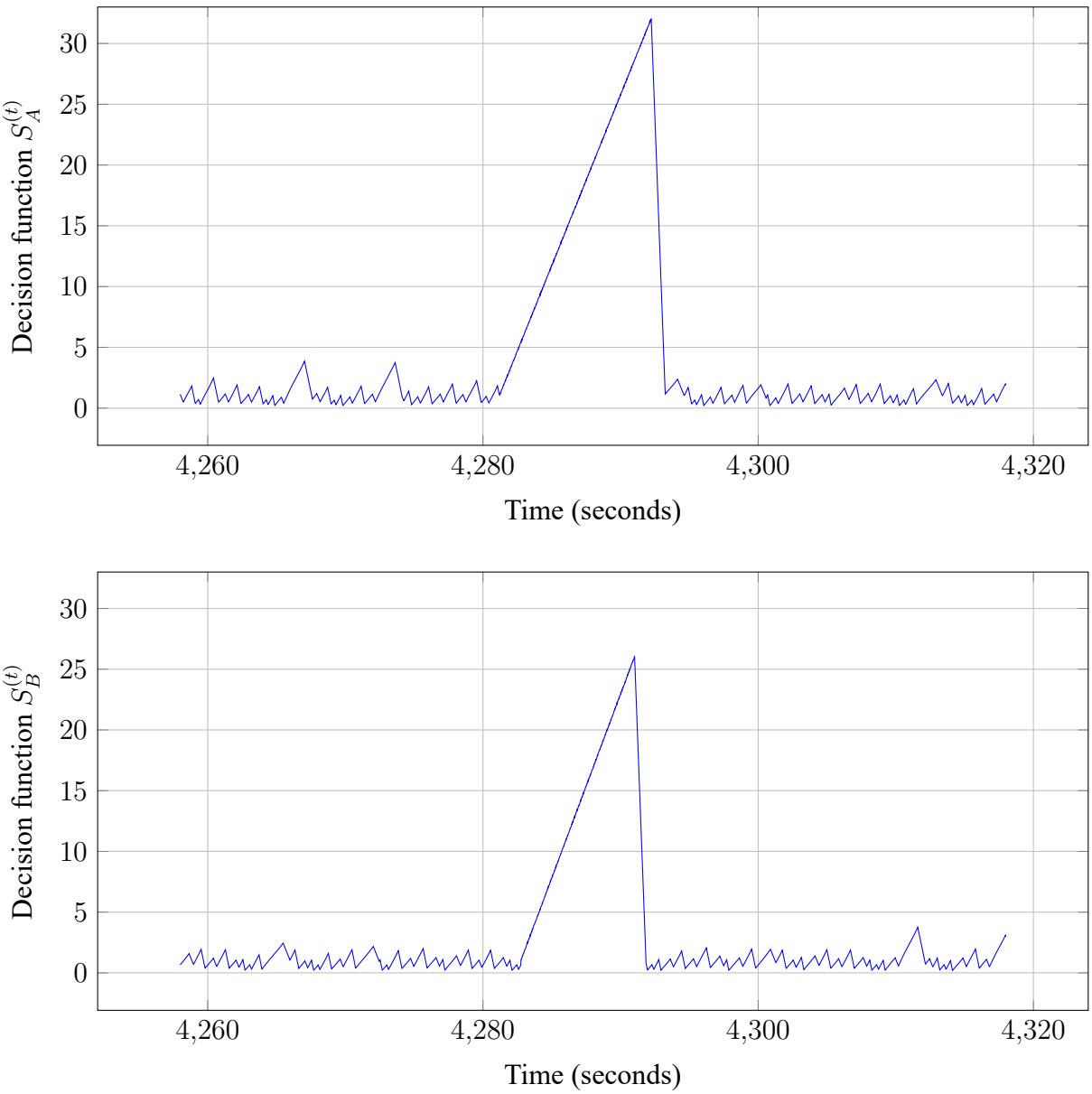


Figure 5.7: Decision functions $S_A^{(t)}$ and $S_B^{(t)}$ for two categorical variables of a test set. The built-in test has reported an anomaly on variable B, and our localisation algorithm has localised the anomaly on two variables. The anomaly impact first the low-level variable A and then appear on variable B.

decision statistics profile such as the one in Figure 5.8. Since these anomalies might occur in the training set, their maximum values will be anomalously high, and so will be the thresholds for detecting anomalies in the test set. To address this issue, we have either added a noise (as for the temperature case explained in section 5.1, or we have removed this starting period from the data.

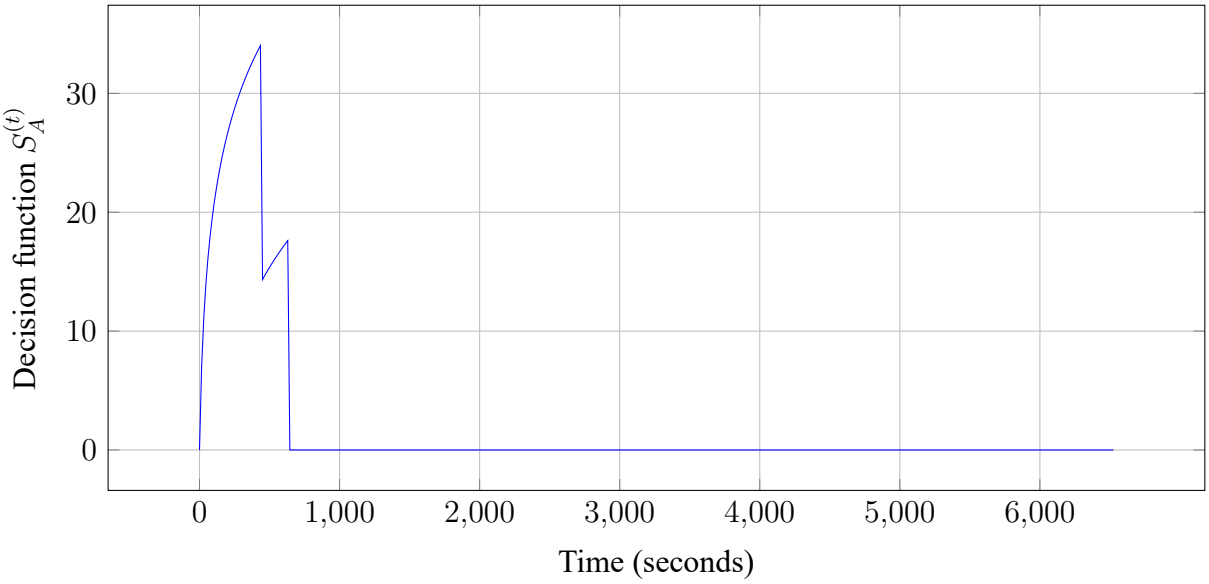


Figure 5.8: Decision functions (4.5) of a variable A which has a starting period causing anomalies.

Finally note that, for a substantial number of cases, the breakdowns lies in the frequencies of anomalies, not in the anomalies themselves; Figure 5.5 depict a standard situation where many anomalies are visible, but for which a breakdown would only occur if these anomalies are too close in time. This kind of breakdowns are not detected by our approach and could fuel further works on anomaly detection.

Chapter 6

Discussion

We presented our anomaly detection and localisation approach to address the improvement of the built-in test of the RBE2. The discussion we made in chapter 2 shows that it is not possible to list all the possible anomalies. We thus focused on an semi-supervised approach, where as a first step data without anomalies were used to learn a model, and as a second step this learned model is exploited to perform anomaly detection of new records.

Built-in test The built-in test allows the radars to evaluate its own working state and detect anomalies. These tasks rely on data that the built-in test is constantly producing and broadcasting on the internal communication bus, and the anomaly detection is performed by filtering the data with rules written by the experts over the years. The built-in test is thus affected by a too high false negative rate and has a limited anomaly localisation ability.

Mixed graphical models Our anomaly detection and localisation algorithm is based on a specific type of pairwise undirected graphical models that we called mixed graphical models. This model is designed for mixing categorical and quantitative variables without any transformation of this variables. Our mixed graphical model framework is built by mixing two classic pairwise undirected models, the Ising model and the Gaussian model, so that the conditional distributions of each variable given the others is either a Bernoulli distribution or a univariate Gaussian distribution depending on the type of each variable.

Mixed model learning We studied two approaches for learning a mixed graphical model from data, all based on the minimisation of a penalised score. If the score we used are always differentiable, we used Lasso and group Lasso regularisations. The optimisation of such regularised scores is done using the proximal gradient algorithm. First, we study the case where the score function is the negative log-likelihood. We showed that the log-likelihood is a concave function of the parameters of the model. However, generally, the likelihood of an undirected model is

intractable, because its partition function has no closed-form and can not be computed. We thus proposed a method to estimate this partition function, and the penalised log-likelihood is finally minimised using a stochastic version of the proximal gradient algorithm. Secondly, we also studied the case where the score function is the log-pseudo-likelihood. The pseudo-likelihood is admittedly sub-optimal, but can be expressed in closed-form and do not require any estimation.

Anomaly detection and localisation In our work, we defined anomalies as samples for which the underlying density has changed. The localisation task, what we defined as the task of finding the variables that caused the anomalies, is performed by detecting a change in the parameters of conditional distribution of each variable given the others. For that purpose, we use a two-sided version of the CUSUM algorithm. We propose a decision statistic, based on conditional log-likelihood ratios of each variable given the others.

All the methods we developed were designed to address the industrial need of Thales of building a complementary tool for the built-in test. They were also designed to match the industrial constraints and challenges imposed by Thales, that is, dealing with heterogeneous variables, anticipating the future embedding in the Rafale for operational use by designing an online anomaly localisation algorithm, and in particular, all the process, from the data processing to the production of the report of the localised anomalies, should need less than 2 hours, which is around the production time of the data. In particular, the detection and localisation task require around 15 minutes for a batch file of 2 million samples of 1000 variables, on a standard technical machine.

There are, however, some assumptions and limitation of our approaches.

- We assumed that the data produced by the built-in test can be modelled by a pairwise model.
- We assumed that, conditionally, the quantitative variables have all an univariate Gaussian distribution, and that, conditionally, the categorical variables have all an Ising model. However, some variables used by the built-in test have already a specific distribution, like counters, which conditional distribution could be modelled with Poisson distribution.

We also outline some questions we did not address:

- the setting of the detection threshold and the sensibility parameter,
- the generalisation of the mixing model to heterogeneous variables with more than two types of variables.

We expect our work to be directly used in the production of radar. So far, with the settings proposed in chapter 5, our algorithms performed well on the test cases, that is, all the anomalies

detected by the built-in tests are also detected by our algorithms, as well as other undetected true anomalies. The localisation of these anomalies did also performed well: the variables localised as causes of the anomalies were always directly related to the physical anomalies, and helped the expert investigating the breakdowns much faster than usually.

Bibliography

Aggarwal, C. C. (2005). On abnormality detection in spuriously populated data streams. In *SDM*, pages 80–91. SIAM. 39

Agyemang, M., Barker, K., and Alhajj, R. (2006). A comprehensive survey of numeric and symbolic outlier mining techniques. *Intelligent Data Analysis*, 10(6):521–538. 39

Atchade, Y. F., Fort, G., and Moulines, E. (2015). On stochastic proximal gradient algorithms. *arXiv preprint arXiv:1402.2365v2*. 21, 58, 78, 81, 82, 84

Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. (2012). Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106. 73, 74, 76

Bach, F. R. and Jordan, M. I. (2002). Learning graphical models with mercer kernels. In *Advances in Neural Information Processing Systems*, pages 1009–1016. 75

Barbu, A. and Zhu, S.-C. (2005). Generalizing swendsen-wang to sampling arbitrary posterior probabilities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1239–1253. 66

Basseville, M., Nikiforov, I. V., et al. (1993). *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs. 14, 15, 58, 100, 102, 109

Bauschke, H. H. and Combettes, P. L. (2011). *Convex analysis and monotone operator theory in Hilbert spaces*. Springer Science & Business Media. 78

BayesiaLab (2013). Bayesia. 49, 91

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202. 9, 79, 80

Becker, S. R., Candès, E. J., and Grant, M. C. (2011). Templates for convex cone problems with applications to sparse signal recovery. *Mathematical programming computation*, 3(3):165–218. 91

Bennett, C. and Campbell, K. (2001). A linear programming approach to novelty detection. *Advances in neural information processing systems*, 13(13):395. 39

Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena scientific Belmont. 74

Besag, J. (1975). Statistical analysis of non-lattice data. *The Statistician*, 24. 10, 11, 58, 76, 85, 100

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. 3, 45, 61

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press. 86

Buntine, W. (1991). Theory refinement on bayesian networks. In *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann Publishers Inc. 49

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15. 13, 14, 39, 97

Cheng, J., Li, T., Levina, E., and Zhu, J. (2013). High-dimensional mixed graphical models. *arXiv preprint arXiv:1304.2810*. 75

Chickering, D., Geiger, D., and Heckerman, D. (1995). Learning bayesian networks: Search methods and experimental results. In *proceedings of fifth conference on artificial intelligence and statistics*, pages 112–128. 49

Combettes, P. L. and Pesquet, J.-C. (2011). Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer. 89

Cooper, G. F. and Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347. 49

d'Aspremont, A., Banerjee, O., and El Ghaoui, L. (2008). First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 30(1):56–66. 74

Davis, D. and Yin, W. (2015). A three-operator splitting scheme and its optimization applications. *arXiv preprint arXiv:1504.01032*. 84

DeGroot, M. H. M. H. et al. (1986). *Probability and statistics*. Number 04; QA273, D4 1986. 45

Dubitzky, W., Granzow, M., and Berrar, D. P. (2007). *Fundamentals of data mining in genomics and proteomics*. Springer Science & Business Media. 40

Duncan, A. (1986). Quality control and industrial statistics, chicago: Richard d. irwin. 101

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441. 75

Friedman, N. and Koller, D. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press. 41, 42, 47, 69, 74, 93

Fu, W. J. (1998). Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistic*, 7(3):397–416. 74

Geiger, D. and Heckerman, D. (1994). Learning gaussian networks. In *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*, pages 235–243. Morgan Kaufmann Publishers Inc. 43, 46

Girshick, M. A. and Rubin, H. (1952). A bayes approach to a quality control model. *The Annals of mathematical statistics*, pages 114–125. 101

Glover, F. and Laguna, M. (2013). *Tabu Search*. Springer. 49

Guttormsson, S. E., Marks, R., El-Sharkawi, M., and Kerszenbaum, I. (1999). Elliptical novelty grouping for on-line short-turn detection of excited running rotors. *IEEE Transactions on Energy Conversion*, 14(1):16–22. 39

Guyon, X. and Künsch, H. R. (1992). Asymptotic comparison of estimators in the ising model. In *Stochastic Models, Statistical Methods, and Algorithms in Image Analysis*, pages 177–198. Springer. 87

Haslbeck, J. and Waldorp, L. J. (2015a). mgm: Structure estimation for time-varying mixed graphical models in high-dimensional data. *arXiv preprint arXiv:1510.06871*. 91

Haslbeck, J. and Waldorp, L. J. (2015b). Structure estimation for mixed graphical models in high-dimensional data. *arXiv preprint arXiv:1510.05677*. 75

Heckerman, D. and Geiger, D. (1995). Learning bayesian networks: a unification for discrete and gaussian domains. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI95)*. 46

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243. 43

Hodge, V. J. and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126. 39

Horn, P. S., Feng, L., Li, Y., and Pesce, A. J. (2001). Effect of outliers and nonhealthy individuals on reference interval estimation. *Clinical Chemistry*, 47(12):2137–2145. 39

Ising, E. (1925). Beitrag zur theorie des ferromagnetismus. *Zeitschrift fur Physik*, 31:253–258. 53

Kemkemian, S., Larroque, A., and Enderli, C. (2013). The industrial challenges of airborne aesa radars. 49, 57, 97

Laby, R., Gramfort, A., Roueff, F., Enderli, C., and Alain, L. (2015). Sparse pairwise Markov model learning for anomaly detection in heterogeneous data. 54, 56, 62, 75, 76

Laurikkala, J., Juhola, M., Kentala, E., Lavrac, N., Miksch, S., and Kavsek, B. (2000). Informal identification of outliers in medical data. In *Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, pages 20–24. 39

Lauritzen, S. L. (1996). *Graphical models*, volume 17. Clarendon Press. 41, 54, 74, 75

Lauritzen, S. L. and Wermuth, N. (1989). Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics*, pages 31–57. 43, 46

Lee, J. D. and Hastie, T. J. (2015). Learning the structure of mixed graphical models. *Journal of Computational and Graphical Statistics*, 24(1):230–253. 11, 20, 54, 56, 62, 75, 76, 85, 90, 91, 94

Lehmann, E. L. and Romano, J. P. (2006). *Testing statistical hypotheses*. Springer Science & Business Media. 47

Lerner, U., Parr, R., Koller, D., Biswas, G., et al. (2000). Bayesian fault detection and diagnosis in dynamic systems. In *AAAI/IAAI*, pages 531–537. 49

Li, H. and Gui, J. (2006). Gradient directed regularization for sparse gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics*, 7(2):302–317. 74

Liang, J., Fadili, J., and Peyré, G. (2014). Local linear convergence of forward–backward under partial smoothness. In *Advances in Neural Information Processing Systems*, pages 1970–1978. 94

Lin, J., Keogh, E., Fu, A., and Van Herle, H. (2005). Approximations to magic: Finding unusual medical time series. In *18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)*, pages 329–334. IEEE. 39

Loh, P.-L., Wainwright, M. J., et al. (2013). Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses. *The Annals of Statistics*, 41(6):3022–3049. 74, 75, 94

Lorden, G. (1971). Procedures for reacting to a change in distribution. *The Annals of Mathematical Statistics*, pages 1897–1908. 101, 109

Lung-Yut-Fong, A., Lévy-Leduc, C., and Cappé, O. (2011). Homogeneity and change-point detection tests for multivariate data using rank statistics. *arXiv preprint arXiv:1107.1971*. 111

Malioutov, D. M., Johnson, J. K., and Willsky, A. S. (2006). Walk-sums and belief propagation in gaussian graphical models. *Journal of Machine Learning Research*, 7(Oct):2031–2064. 54

Margaritis, D. (2003). *Learning Bayesian network model structure from data*. PhD thesis, US Army. 47

Markou, M. and Singh, S. (2003a). Novelty detection: a review—part 1: statistical approaches. *Signal processing*, 83(12):2481–2497. 97

Markou, M. and Singh, S. (2003b). Novelty detection: a review—part 2:: neural network based approaches. *Signal processing*, 83(12):2499–2521. 97

Mase, S. (1995). Consistency of the maximum pseudo-likelihood estimator of continuous state space gibbsian processes. *The Annals of Applied Probability*, pages 603–612. 87

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs ad variable selection with the lasso. *The annals of statistics*, pages 1436–1462. 75, 76

Moustakides, G. V. (1986). Optimal stopping times for detecting changes in distributions. *The Annals of Statistics*, pages 1379–1387. 109

Ng, A. Y. (2004). Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78. ACM. 6, 72

Nigam, K., Lafferty, J., and McCallum, A. (1999). Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67. 71

Page, E. (1954). Continuous inspection schemes. *Biometrika*, 41(1/2):100–115. 14, 58, 101, 109

Parikh, N. and Boyd, S. (2013). Proximal algorithms. *Foundations and Trends in optimization*, 1(3):123–231. 8, 9, 76, 77, 79, 80, 84

Patcha, A. and Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer networks*, 51(12):3448–3470. 39

Pearl, J. (2014). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann. 42

Potts, R. (1953). Some generalized order-disorder transformations. *Proc. Cambridge Philosophie Soc*. 3, 54

Raguet, H., Fadili, J., and Peyré, G. (2013). A generalized forward-backward splitting. *SIAM Journal on Imaging Sciences*, 6(3):1199–1226. 9, 78, 83

Rashidi, L., Hashemi, S., and Hamzeh, A. (2011). Anomaly detection in categorical datasets using bayesian networks. In *Artificial Intelligence and Computational Intelligence*, pages 610–619. Springer. 49

Ravikumar, P., Wainwright, M. J., Lafferty, J. D., et al. (2010). High-dimensional ising model selection using ℓ_1-regularized logistic regression. *The annals of statistics*, 38(3):1287–1319. 75, 76

Ritov, Y. (1990). Decision theoretic optimality of the cusum procedure. *The Annals of Statistics*, pages 1464–1469. 109

Roberts, S. (2000). Control chart tests based on geometric moving averages. *Technometrics*, 42(1):97–101. 101

Sakurada, M. and Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, page 4. ACM. 39

Schmidt, M. (2010). *Graphical model structure learning with l1-regularization*. PhD thesis, University Of Bristish Columbia (Vancouver). 3, 57, 61, 72, 74

Schmidt, M. W., Murphy, K. P., Fung, G., and Rosales, R. (2008). Structure learning in random fields for heart motion abnormality detection. In *CVPR*, volume 1, page 2. 74

Shachter, R. D. and Kenley, C. R. (1989). Gaussian influence diagrams. *Management science*, 35(5):527–550. 46, 47

Shewhart, W. A. (1931). *Economic control of quality of manufactured product*. ASQ Quality Press. 101

Shiryaev, A. (1961). The problem of the most rapid detection of a disturbance in a stationary process. In *Soviet Math. Dokl*, volume 2. 101

Song, X., Wu, M., Jermaine, C., and Ranka, S. (2007). Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 19(5):631–645. 39, 40

Speed, T. and Kiiveri, H. (1986). Gaussian markov distributions over finite graphs. *The Annals of Statistics*, pages 138–150. 54

Spiegelhalter, D. J. and Lauritzen, S. L. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20(5):579–605. 45

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288. 6, 71, 75

Tur, I. and Castelo, R. (2012). Learning mixed graphical models from data when p is larger than n. *arXiv preprint arXiv:1202.3765*. 75

udea Pearl, T. V. J. (1991). Equivalence and synthesis of causal models. In *Proceedings of Sixth Conference on Uncertainty in Artijicial Intelligence*, pages 220–227. 47

Valizadegan, H. and Tan, P.-N. (2007). Kernel based detection of mislabeled training examples. In *SDM*, pages 309–319. SIAM. 39

Valko, M. (2011). *Adaptive Graph-Based Algorithms for Conditional Anomaly Detection and Semi-Supervised Learning*. PhD thesis, University of Pittsburgh. 39, 40, 97, 98

Valko, M., Cooper, G. F., Seybert, A., Visweswaran, S., Saul, M., and Hauskrecht, M. (2008). Conditional anomaly detection methods for patient-management alert systems. In *Workshop on Machine Learning in Health Care Applications in The 25th International Conference on Machine Learning*, Helsinki, Finland. 39, 40

Valko, M., Kveton, B., Valizadegan, H., Cooper, G. F., and Hauskrecht, M. (2011). Conditional anomaly detection with soft harmonic functions. In *2011 IEEE 11th International Conference on Data Mining*, pages 735–743. IEEE. 14, 40, 98

Varoquaux, G., Gramfort, A., Poline, J.-B., and Thirion, B. (2010). Brain covariance selection: better individual functional connectivity models using population prior. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *NIPS*, pages 2334–2342. Curran Associates, Inc. 72

Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305. 53, 74

Wainwright, M. J., Lafferty, J. D., and Ravikumar, P. K. (2006). High-dimensional graphical model selection using ℓ_1-regularized logistic regression. In *Advances in neural information processing systems*, pages 1465–1472. 74, 92

Wald, A. (1973). *Sequential analysis*. Courier Corporation. 101

Whittaker, J. (2009). *Graphical models in applied multivariate statistics*. Wiley Publishing. 41

Wolff, U. (1989). Collective monte carlo updating for spin systems. *Physical Review Letters*, 62(4):361. 66

Wong, W.-K., Moore, A., Cooper, G., and Wagner, M. (2003). Bayesian network anomaly pattern detection for disease outbreaks. In *ICML*, pages 808–815. 39, 49

Yang, E., Allen, G., Liu, Z., and Ravikumar, P. K. (2012). Graphical models via generalized linear models. In *Advances in Neural Information Processing Systems*, pages 1358–1366. 76

Yang, E., Baker, Y., Ravikumar, P. D., Allen, G. I., and Liu, Z. (2014). Mixed graphical models via exponential families. In *AISTATS*, pages 1042–1050. 55, 56

Yang, E., Ravikumar, P., Allen, G. I., and Liu, Z. (2015). Graphical models via univariate exponential family distributions. *Journal of Machine Learning Research*, 16:3813–3847. 54, 62

Yang, E., Ravikumar, P. K., Allen, G. I., and Liu, Z. (2013). Conditional random fields via univariate exponential families. In *Advances in Neural Information Processing Systems*, pages 683–691. 56, 76

Ye, N. and Xu, M. (2000). Probabilistic networks with undirected links for anomaly detection. *IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, pages 175–179. 49

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67. 6, 72

Index

Détection et localisation d'anomalies par utilisation de modèles graphiques mixtes

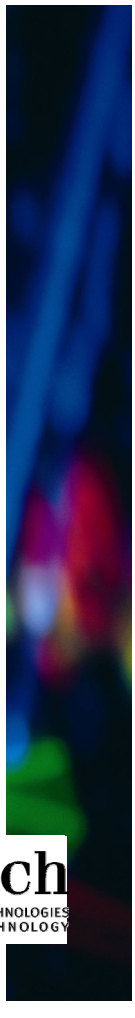
Romain Laby

RESUME : Cette thèse s'articule autour d'un besoin industriel de la société Thales Système Aéroportés et du radar de combat RBE2 équipant les avions de chasses Dassault Rafale. Elle développe une méthodologie de localisation d'anomalies dans des flux de données hétérogènes en utilisant un modèle graphique mixte non orienté et pairs à pairs. Les données sont un mélange de variables catégorielles et quantitatives, et le modèle est appris à partir d'un jeu de données dont on suppose qu'il ne contient pas de données anormales. Les algorithmes de localisation d'anomalies utilisent une version adaptée de l'algorithme CUSUM, dont la fonction de décision est basée sur le calcul de ratios de vraisemblance conditionnelles. Cette fonction permet de réaliser une détection d'anomalies variable par variable et de localiser précisément les variables impliquées dans l'anomalie.

MOTS-CLEFS : Détection d'anomalies, modèles graphiques, données hétérogènes, flux de données

ABSTRACT : This thesis revolves around an industrial need of Thales Système Aéroportés and the RBE2 combat radar equipping Dassault Rafale fighter aircraft. It develops a methodology for locating anomalies in heterogeneous data stream using a mixed, non-orientation and peer-to-peer graphical model. The data are a mixture of categorical and quantitative variables, and the model is learned from a data set that is assumed not to contain abnormal data. Anomaly localization algorithms use an adapted version of the CUSUM algorithm, whose decision function is based on the calculation of conditional likelihood ratios. This function allows the detection of variable anomalies per variable and the precise localization of the variables involved in the anomaly.

KEY-WORDS : Anomaly detection, graphical models, heterogeneous data, data stream



TELECOM ParisTech

EDITE

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY