UNIVERSITÉ DE PARIS 13,
SORBONNE PARIS CITÉ
ÉCOLE DOCTORALE GALILÉE

THÈSE

présentée par

Amine CHAIBI

pour obtenir le grade de
DOCTEUR D'UNIVERSITÉ
SPÉCIALITÉ : INFORMATIQUE

Contribution en apprentissage topologique non supervisé pour la fouille de données

soutenue publiquement le 29 novembre 2013

Membre du jury :

Directeur :	Mustapha LEBBAH (HDR)	-	LIPN, Université Paris 13
Co-encadrement :	Hanane AZZAG (HDR)	-	LIPN, Université Paris 13
Rapporteurs :	Pascale KUNTZ (Pr)	-	LINA, Polytech Nantes
	Christel VRAIN (Pr)	-	LIFO, Université d'Orléans
Examinateurs :	Gilles BISSON (CR CNRS)	-	LIG, Université de Grenoble
	Ndeye Niang KEITA (MCF)	-	Cedric, Cnam
	Mohamed NADIF (Pr)	-	LIPADE, Université Paris 5
	Céline ROUVEIROL (Pr)	-	LIPN, Université Paris 13
Invité :	Richard DOMPS (PDG)	-	Anticipeo, Paris

Table des matières

Table des figures

Liste des tableaux

Résumé :

Le travail de recherche exposé dans cette thèse concerne le développement d'approches à base des cartes auto-organisatrices pour les problèmes de détection de groupes-outliers et de nouveautés, de bi-partitionnement, ainsi que l'estimation des intervalles de confiance des prévisions de la société Anticipeo. Pour chaque problématique, un modèle d'apprentissage non supervisé adapté est proposé. La première contribution de cette thèse est dédiée à la détection de groupes-outliers en proposant une nouvelle mesure nommée GOF (Group Outlier Factor), qui est estimée par l'apprentissage non supervisé. Nous l'avons intégré dans l'apprentissage des cartes topologiques. Notre approche est basée sur la densité relative de chaque groupe de données. Elle fournit simultanément un partitionnement des données et un indicateur quantitatif (GOF) sur "la particularité" de chaque cluster ou groupe de données. Par la suite, la mesure GOF est utilisée comme classifieur pour la détection de nouveautés. En effet, nous développons une approche s'appuyant sur le GOF qui permet de détecter automatiquement les données nouvelles qui n'étaient pas connues au moment de l'apprentissage.

La seconde contribution concerne le problème de bi-partitionnement (bi-clustering). L'approche que nous développons, qui se nomme BiTM (Bi-clustering using Topological Map), permet de représenter simultanément dans une carte topologique les observations et les variables d'une matrice de données. Contrairement à certaines approches de l'état de l'art, BiTM ne nécessite aucune pré-organisation de la matrice de données. Notre approche permet aussi de fournir de nouvelles visualisations.

Enfin, la troisième contribution, qui est de caractère applicatif, aborde le problème d'estimation des intervalles de confiance dans les séries chronologiques. La société Anticipeo propose une solution informatique qui permet de réaliser des prévisions détaillées des ventes pour différents clients. En supplément de son offre standard, nous avons développé une offre complémentaire d'estimation d'intervalles de confiance ("marges d'erreur") et de la classification des produits selon leurs caractéristiques statistiques.

Les différentes évaluations réalisées dans cette thèse (mesures de performances et visualisations) ont obtenu des résultats intéressants.

Mots clés : apprentissage non supervisé, clustering, bi-clustering, cartes SOM, outliers, groupes-outliers, détection de nouveautés, estimation des intervalles de confiance.

Abstract :

The research outlined in this thesis concern the development of approaches based on self-organizing maps for the groups-outliers and novelty detection, bi-clustering and confidence intervals estimation. For each problem, an unsupervised learning model is proposed. The first model that we propose in this thesis is dedicated to groups-outliers detection by proposing a new measure nammed GOF (Group Outlier Factor), which is estimated by the unsupervised learning. We integrated it to topological maps learning. Our approach is based on the density of each group of data, and simultaneously provides a data partitioning and a quantitative indicator (GOF) that indicat the "outlier-ness" of each cluster or group. Thereafter, the GOF measure is used as a classifier for novelty detection problem. In fact, we develop an approach based on GOF which automatically detects the new data that were not known during the learning process.

The second model developed in this thesis is related to bi-clustering problem titled BiTM (Bi-clustering using Topological Map). BiTM is based on self-organizing maps and provides a simultaneous clustering of rows and columns of the data matrix in order to increase the homogeneity of bi-clusters by respecting neighborhood relationship and using a single map. BiTM maps provide a new topological visualization of the bi-clusters.

The third contribution is addressed to the confidence intervals estimation problem in time series. The Anticipeo company offers a solution that allows to perform detailed forecasts for different customers. In addition to its standard solution, we have developed a complementary tool for confidence intervals estimation and products classification according to their statistical characteristics.

In this thesis, we have used different evaluation using performance measure and visualizations. The obtained results are encouraging and promising to continu in this direction.

Keywords : unsupervised learning, clustering, bi-clustering, self-organizing maps, outliers, groups outliers, novelty detection, confidence intervals estimation.

Remerciements

Je voudrais tout d'abord remercier grandement mes encadrants Hanane Azzag et Mustapha Lebbah, pour leur constante disponibilité et leurs précieux conseils, qui ont permis à ce travail de voir le jour. Je suis ravi d'avoir travaillé en leur compagnie car outre leur appui scientifique, ils ont toujours été présent pour me soutenir et me conseiller durant ces trois années. Je suis extrêmement sensible à leurs qualités humaines et à leur sens de l'écoute et de compréhension.

Je remercie également madame Pascale Kuntz et madame Christel Vrain qui m'ont fait l'honneur d'être rapporteurs de cette thèse. J'ai apprécié le regard critique qu'elles ont porté à mon travail ainsi que les remarques pertinentes proposées pour améliorer ce manuscrit. Je remercie aussi les autres membres du jury : Madame Ndeye Niang Keita, Monsieur Gilles Bisson et Monsieur Mohamed Nadif d'avoir accepté de faire parti de ce jury.

J'aimerais également remercier Monsieur Richard Domps, PDG de la société Anticipeo pour avoir financé et suivi mes travaux de thèse. Ses conseils et remarques, toujours pertinents, m'ont permis de confronter et ajuster mes travaux à des problématiques pratiques et réelles, sans oublier la confiance et l'intérêt qu'il a manifesté pour ce travail.

Durant ces trois ans, j'ai travaillé dans un cadre particulièrement agréable, grâce à l'ensemble des membres du laboratoire d'informatique de Paris nord (LIPN). Je remercie particulièrement la directrice du laboratoire Madame Laure Petrucci, notre responsable d'équipe A3 Madame Céline Rouveirol. Je remercie Monsieur Faouzi Boufares (HDR) dont la gentillesse n'a pas d'égale, Monsieur Kais Klai (MCF), qui anime efficacement la vie du laboratoire et Monsieur Lazhar Labiode (MCF), pour l'intérêt dont il a fait preuve envers ma recherche.

Ces remerciements seraient incomplets si je n'en adressais pas aux secrétaires du LIPN pour leur soutien logistique et moral. Je pense plus particulièrement à Madame Nathalie Tavares et Madame Brigitte Guéveneux. Il est difficile de trouver des qualificatifs assez forts pour décrire leur sympathie et leur gentillesse.

La réussite humaine d'une thèse dépend en grande partie de ceux que l'on côtoie quotidiennement, à savoir les doctorants. Un grand merci à : Hanane

Allaoua, Aicha Bensalem, Hanane Bouzid Tafat, Ines Chebil, Karima Mouhoubi, Hanene Ochi, Manisha Pujari, Naim Aber, Alois Brunel, Nhat-Quang Doan, Ehab Hassan, Mohamed Hindaoui, Tugdual Sarazin et Zeid Yakoubi. Je tiens aussi à remercier tous les membres de l'association des doctorants de Galilée (ADG) pour tous les moments de plaisir que nous avons pu passer ensemble au cours de ces trois dernières années.

Ma reconnaissance va à ceux qui ont plus particulièrement assuré le soutien moral de cette thèse, qu'ils se trouvent en France ou de l'autre côté de la méditerranée (Algérie) : mes frères et soeurs Lynda, Nassima, Azzedine et Sofiane. Mes cousines et cousins Lamia, Sabrina, Amir, Faouzi, Hakim, Khlifa, Kiki, Kouceila, Lotfi, Redouane et Youba. Mes amis Julia, Joa, Louiza, Marie, Maya, Meriem, Moricette, Sarah, Valantina, Victoria, Adel, Aniss, Aziz, Bilal, Djafar, Esteban, Kacim, Khalil, Mourad, Thibault, Thomas et Yacine. Merci à toutes et à tous de m'avoir aidé, encouragé et contribué au maintien de mon moral. Un merci particulier à Célia Faïd, d'avoir vécu avec moi cette importante étape durant un long parcours commencé il y a bien longtemps avec beaucoup d'amour et d'affection...

Enfin, les mots les plus simples étant les plus forts, j'adresse tout mon amour à ma mère et à mon père qui m'ont poussé à la recherche du meilleur de moi en m'investissant à fond pour explorer mes limites et atteindre la perfection. Malgré mon éloignement depuis de nombreuses années, leur tendresse et amour me portent et me guident tous les jours. Merci d'avoir fait de moi ce que je suis devenu aujourd'hui. Est-ce un bon endroit pour dire ce genre de choses ? Je n'en connais pas en tous cas un mauvais...

Liste des publications

Journal avec comités de lecture (1 article)

Amine Chaibi, Mustapha Lebbah and Hanane Azzag, Group-Outlier Factor : a new score using Self-Organising Map for Group-Outlier and Novelty Detection International Journal of Computational Intelligence and Applications *(IJCIA)*, World Scientific Publishing Company.

Conférences internationales avec comités de lecture (4 articles)

Amine Chaibi, Mustapha Lebbah and Hanane Azzag. A New Bi-clustering Approach Using Topological Maps. International Joint Conference on Neural Networks *(IJCNN'13)* . August 4-9, 2013, Dallas, USA.

Amine Chaibi, Mustapha Lebbah and Hanane Azzag. A new visualization of group-outliers in unsupervised learning. 17th International Conference Information Visualisation *(IV'13)*. 15 - July 2013, London, UK.

Amine Chaibi, Mustapha Lebbah and Hanane Azzag. Novelty Detection using a New Group Outlier Factor. 19th International Conference on Neural Information Processing *(ICONIP'12)*. Regular session, Part III, LNCS 7665, p. 364-372. November 12-15 2012, Doha Qatar.

Amine Chaibi, Hanane Azzag and Mustapha Lebbah. Automatic Group-Outlier Detection. European Symposium on Artificial Neural Networks Computational Intelligence and Machine Learning *(ESANN'12)*. Bruges, Belgium, 25 - 27 April 2012, pp. 393-398.

Conférences nationales avec comités de lecture (2 articles)

Amine Chaibi, Mustapha Lebbah and Hanane Azzag. Nouvelle approche de bi-partitionnement topologique. 29 janvier - 01 février 2013, Toulouse, France, pp. 67-78. RNTI, Revue des Nouvelles Technologies de l'Information, Editions Hermann *(EGC'13)*.

Amine Chaibi, Mustapha Lebbah and Hanane Azzag. Détection de groupes outliers en classification non supervisée. 31 janvier - 3 février, Bordeaux, France, pp. 119-125 RNTI, Revue des Nouvelles Technologies de l'Information, Editions Hermann *(EGC'12)*.

Communications orales (1 présentation et 1 poster)

Amine Chaibi, Machine Learning Summer School *(MLSS'12)*, Santa Cruz, California, United States. July 9-20, 2012.

Amine Chaibi, Les journées Big Data *(JBD)*, Tours, France, 18 et 19 juin 2012.

Stage de master (1 article)

Amine Chaibi, Nacima Labadie and Christian Prins. Bicriteria Obnoxious Facility Location : Matheuristic Approach. The Second International Symposium on Operational Research *(ISOR'11)*, Algiers, Algeria. May 30th - June 02nd, 2011. 2 pages.

Introduction générale

L'entreprise Anticipeo et son offre

Anticipeo [1] est une jeune entreprise innovante qui a développé une offre de service et de logiciels qui permet de construire aisément des budgets opérationnels détaillés et fiables, puis d'en suivre précisément l'exécution. À partir d'extraction du système d'information de l'entreprise, les statisticiens d'Anticipeo établissent des projections détaillées de ventes en quantités, valeurs et prix, selon les axes produits et clients. Ces projections sont délivrées à l'entreprise cliente, qui les transforme en budgets détaillés en fonction de ses anticipations métier et de ses plans d'actions.

Contexte et problématiques

Les outils statistiques développés par Anticipeo sont exploités commercialement et fournissent un bon niveau de résultat dans les cas standards (ventes assez récurrentes). Pour accroître le développement commercial de son offre, Anticipeo cherche à étendre son offre dans trois directions :

- Fournir à ses clients des intervalles de confiance sur les projections qui leur sont délivrées,
- Détecter les outliers, les groupes-outliers et les nouveautés,
- Définir des classes de produits et clients homogènes en utilisant la classification non supervisée.

En outre, Anticipeo souhaite améliorer ses méthodes internes d'élaboration des projections pour gagner en efficacité dans le processus de paramétrage et de validation humaine des projections. En effet, ceux-ci font encore largement appel à des processus d'inspection systématique des résultats et sont assez lourds en temps de travail. Pour faciliter l'élaboration des prévisions au niveau le plus fin, différents modèles ont été proposés dans la littérature scientifique. La question fondamentale soulevée par ces modèles est de savoir comment raffiner au maximum les résultats obtenus. Étant donné le niveau élevé qu'exigent les entreprises en matière de fiabilité des prévisions, il sera décisif de leur fournir des intervalles de confiance des projections élaborées, afin de les guider dans l'élaboration de leurs prévisions. Souvent, les données traitées sont représentées d'une manière complexe et multidimensionnelle. C'est pour cela qu'on fait appel aux techniques de fouille de données et d'apprentissage automatique afin de bien les représenter et surtout de mieux les comprendre. Il existe une variété de modèles d'apprentissage dans la littérature scientifique.

1. http ://anticipeo.fr/

Ils varient selon le domaine d'application ou l'algorithme d'apprentissage utilisé. Nous pouvons les caractériser en plusieurs dimensions :

- Paradigme d'apprentissage (supervisé, non supervisé, par renforcement),
- Paradigme d'algorithme (réseaux de neurones, bio-inspirés, règles symboliques),
- Mode d'implémentation,
- Environnement déterministe/probabiliste, markovien/stochastique.

L'ensemble des travaux de cette thèse se situent dans la première famille de modèle, c'est-à-dire le paradigme d'apprentissage.

Contributions

Dans le cadre de cette thèse, nous proposons de développer des outils de fouille de données à base de techniques d'apprentissage pour l'exploration des données. Les techniques (algorithmes et protocoles d'analyse) utilisées actuellement pour l'analyse de données sont des algorithmes de segmentation, décomposition factorielle, modèles graphiques et réseaux de neurones dynamiques. Les résultats obtenus par la mise en œuvre de ces outils sont généralement de bonne qualité, mais présentent les limites suivantes :

- Les algorithmes de segmentation demandent un travail important de mise en place pour une durée de vie parfois très courte,
- Le résultat obtenu dépend pour beaucoup de la quantité et de la qualité des données,
- Les approches proposées ne prennent pas en compte toutes les dimensions du problème de la connaissance métier.

Un des axes de travail de cette thèse est celui de la segmentation. Cela permet à Anticipeo de fournir des conseils aux entreprises, mais également de classer les populations clients, produits et couples clients-produits en segments homogènes sous l'angle du comportement statistique (tant sur le plan de l'évolution passée et prévisionnelle que de celui des écarts entre prévu et réalisé) afin de traiter plus finement ces populations sans devoir inspecter une à une les nombreuses occurrences, et également de traiter des séries statistiques moins récurrentes. Notre démarche pour l'analyse de données dans cette thèse s'articule autour de l'exploration et l'extraction de connaissances à partir des données.

Dans le cadre de cette thèse (convention CIFRE entre Anticipeo et le Laboratoire d'Informatique de Paris Nord LIPN, UMR 7030-CNRS), nous proposons d'étudier un ensemble de problématiques des modèles d'apprentissage non supervisé et leur mise en œuvre avec lesquels nous offrons des avantages et des spécificités par rapport aux autres techniques. Cela nous a conduit à réaliser des études ponctuelles sur des points très ciblés dans ce do-

maine, qui permettent de détecter automatiquement les "groupes-outliers" et les nouveautés, de regrouper les individus et les variables les plus homogènes dans des bi-classes (bi-clusters), l'estimation des intervalles de confiance et la classification des produits Anticipeo. Nous avons exploré principalement dans cette thèse les axes suivants :

1. **Détection des groupes-outliers et des nouveautés**
   Un outlier (point aberrant) est une donnée ou une observation qui est considérablement différente, divergente, dissemblable ou distincte du reste des données. Il existe une littérature abondante sur les problèmes de détection des "groupes-outliers". Un groupe-outlier est un petit ensemble de données qui ne se comporte pas de la même manière que le reste des données. Souvent, ces groupes sont denses et significativement isolés par rapport aux autres clusters de la base de données. Les outliers ou les groupes-outliers sont problématiques, car ils peuvent biaiser les résultats, notamment pour les méthodes basées sur des distances entre individus. Dans un processus de fouille de données, la détection et le traitement des outliers sont incontournables lors de la préparation des données, ou même après, pour analyser et valider les résultats. Nous proposons dans cette première famille de contribution une approche qui permet de détecter des groupes-outliers. Cette méthode se base sur la densité des données et permet de qualifier la "particularité" de chaque groupe/cluster. Nous proposons un nouveau score, que nous appelons GOF (Group Outlier Factor), qui est intégré dans un processus d'apprentissage non supervisé. À cet effet, nous avons utilisé les cartes auto-organisatrices (SOM) comme algorithme d'apprentissage. Par la suite, nous l'avons utilisé comme classifieur afin de traiter le problème de détection des nouveautés.

2. **Le bi-partitionnement topologique**
   Cet axe concerne le développement de modèles d'apprentissage statistique non supervisé visant à créer simultanément par apprentissage des groupes homogènes (ou typologies) des données et des variables. Ainsi, ces modèles permettront de découvrir un espace topologique d'un ensemble de données et de variables. Cet espace préserve la notion de voisinage entre les données. Il s'agit d'une problématique classique en fouille de données, mais qui demeure intéressante de par les nombreuses applications qui la nécessitent. Dans cette optique, nous proposons dans cette deuxième famille de contribution une approche nouvelle de bi-partitionnement topologique à base des cartes SOM. L'approche est basée sur une nouvelle fonction de coût à minimiser pour aborder cette problématique avec un outil de visualisation et de compréhension des

données à analyser.

3. **Estimation des intervalles de confiance et classification des produits Anticipeo**
   La notion d'intervalle de confiance sur les projections que la société Anticipeo délivre à ses clients est un point essentiel, car nous travaillons des séquences modélisant un historique et nous souhaitons savoir si les prévisions renseignées pas les utilisateurs se situent dans la marge d'incertitude statistique des projections fournies par Anticipeo. C'est pour cela qu'une grande partie de cette thèse a été consacrée à cette problématique. En nous basons sur le modèle de prévision d'Anticipeo, nous avons développé une méthode empirique qui permet de réaliser l'estimation des intervalles de confiance basée sur un calcul des écarts quadratiques moyens sous des hypothèses sur le comportement futur des intervalles. Nous avons, par la suite, construit une base de données en utilisant les données Anticipeo. Cette base de données est utilisée pour la classification automatique des produits Anticipeo, et l'application des modèles de la détection des groupes-outliers et des nouveautés et le bi-partitionnement topologique sur ces données.
   Pour des raisons de confidentialité, nous présentons ni le modèle de prévision d'Anticipeo ni le modèle et l'algorithme de l'approche d'estimation des intervalles de confiance. Nous montrons seulement quelques résultats de cette approche et ceux de la classification des produits Anticipeo.

Organisation de la thèse

Ce manuscrit est organisé en cinq chapitres principaux et une annexe :

- **Chapitre 1 :** ce chapitre est consacré à un état de l'art sur les principales méthodes de clustering et de bi-clustering.
- **Chapitre 2 :** ce chapitre est dédié à l'état de l'art sur les différentes approches de détection d'outliers, de groupes-outliers et des nouveautés.
- **Chapitre 3 :** ce chapitre présente la première contribution de cette thèse sur la détection de groupes-outliers et des nouveautés en utilisant les cartes topologiques.
- **Chapitre 4 :** dans ce chapitre, nous exposons notre deuxième contribution sur le bi-partitionnement topologique.
- **Chapitre 5 :** c'est le dernier chapitre de cette thèse, qui est dédié à l'estimation des intervalles de confiance et à la classification des produits d'Anticipeo.
- **Annexe 1 :** nous exposons dans cette annexe un survol bibliographique sur les méthodes de régression linéaire.

Chapitre 1

État de l'art sur la fouille de données : clustering et bi-clustering

Sommaire

1.1 Introduction

La fouille de données (ou datamining) consiste à rechercher et à extraire de l'information, utile et inconnue, à partir de grands volumes de données

stockées dans des bases ou des entrepôts de données. Le développement récent de la fouille de données (depuis le début des années 1990) est lié à plusieurs facteurs : une puissance de calcul importante est disponible sur les ordinateurs, le volume des bases de données augmente énormément, l'accès aux réseaux de taille mondiale, ces réseaux ayant un débit sans cesse croissant, qui rendent le calcul distribué et la distribution d'information sur échelle mondiale variable. La fouille de données a aujourd'hui une grande importance économique du fait qu'elle permet d'optimiser la gestion des ressources humaines et matérielles. La classification est la tâche la plus importante de la fouille de données et consiste à examiner des caractéristiques d'une observation afin de l'affecter à une classe d'un ensemble donné [Saporta 2006].

1.2 Fouille de données

La fouille de données consiste à parcourir d'immenses volumes de données contenues dans une base de données, à la recherche de connaissances [Tan 2005]. C'est une discipline qui se situe à l'intersection de différents domaines tels que l'informatique, l'intelligence artificielle, l'analyse de données, les statistiques, la théorie des probabilités, l'optimisation, la reconnaissance de formes, les bases de données et l'interaction homme-machine, etc.

L'extraction des connaissances à partir des données (ECD) permet d'obtenir des informations pertinentes à partir de données sur lesquelles nous ne faisons aucune hypothèse, et, de celles-ci, de tirer des connaissances. Fayyad [Fayyad 1996] donne une définition de l'ECD, que la communauté scientifique francophone traduit de la manière suivante : l'ECD est le processus non trivial, interactif et itératif qui permet d'identifier des modèles valides, nouveaux, potentiellement utiles et compréhensibles à partir de bases de données massives. Le terme "processus" signifie que l'ECD se décompose en plusieurs opérations. Ce processus peut être regroupé en cinq phases majeures [Hastie 2009] :

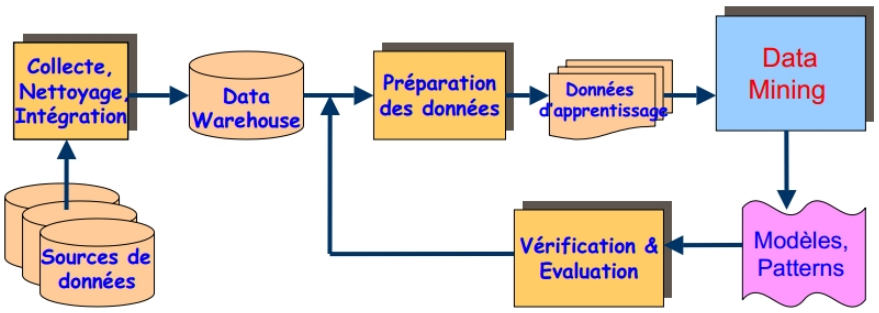


FIGURE 1.1 – Différentes étapes du processus ECD.

1. Compréhension du domaine étudié : lors de cette phase, une analyse du problème et des contraintes qui lui sont attachées doit permettre la collecte de données brutes. Ces données se composent d'observations (ou objets ou individus ou patterns) et des variables qui leur sont associées et qui doivent permettre de décrire au mieux le problème traité. L'utilisateur ne sait pas encore si les données qu'il a réunies seront toutes adaptées à son problème ni si ces données seront suffisantes. Nous sommes en présence des données initiales.

2. Prétraitement : lors de cette phase, un prétraitement est effectué à la fois sur les individus et sur les variables. Cette phase de prétraitement consiste à nettoyer les données, les mettre en forme, traiter les données manquantes, détecter les outliers et les nouveautés, échantillonner les individus, sélectionner et construire des variables. On obtient ainsi un ensemble de données cibles. Cette phase a une place importante au sein du processus d'ECD, car c'est elle qui va déterminer la qualité des modèles construits lors de la phase de fouille de données. Elle peut prendre jusqu'à 60 % du temps dédié au processus d'ECD.

3. Fouille de données : cette phase intègre le choix de la méthode d'apprentissage. Ces choix doivent tenir compte des contraintes liées au domaine étudié ainsi que des connaissances que les experts du domaine peuvent nous fournir. L'algorithme sélectionné est alors appliqué aux données cibles dans le but de rechercher les structures sous-jacentes des données et de créer des modèles explicatifs ou prédictifs. Certes, la fouille de données n'est qu'une étape du processus de l'ECD, mais elle est sans conteste le cœur et le moteur de tout ce processus.

4. Post-traitement : cette phase consiste en l'évaluation et la validation des modèles construits lors de la phase précédente. Ce n'est qu'après cette

phase que les données et l'information tirée deviennent des connaissances.

5. Interprétation et exploitation des résultats : l'interprétation des résultats qui sont sous forme de modèles ou de règles permet d'obtenir des connaissances. Ce sont ces connaissances qui seront fournies à l'utilisateur.

La finalité de l'ECD est de pouvoir traiter des données brutes et volumineuses, et, à partir de ces données, d'établir des connaissances directement utilisables par un expert ou un non-expert du domaine étudié. Les techniques d'ECD deviennent de plus en plus prisées au sein du monde industriel. En effet, les promesses de l'ECD en termes de valorisation de l'information ne peuvent laisser insensibles les acteurs industriels. Tout d'abord parce que l'information apparaît, de nos jours, comme un élément stratégique déterminant. Ensuite, parce que les avancées technologiques en informatique permettent d'augmenter les capacités de stockage et de calcul. Ainsi, si l'on considère comme exemple l'ensemble des tickets de caisse d'un supermarché sur une période de 10 ans, il est aisé d'imaginer la quantité de données présentes, la diversité des caractéristiques, et donc la difficulté conséquente d'une exploitation de l'information présente. Pourtant, on dispose là d'une immense source d'informations, à savoir une quantité suffisamment importante de données pour établir une classification pertinente de la clientèle ainsi que son comportement typique. Le processus d'ECD résout de manière efficace ces difficultés et fournit les connaissances attendues.

1.2.1 Tâches de la fouille de données

Le choix des techniques de fouille de données applicables dépend de la tâche particulière à accomplir et des données disponibles pour l'analyse. La première étape consiste à traduire un objectif en une ou plusieurs tâches. Les principales tâches de fouille de données sont :

- La classification supervisée : consiste à examiner des caractéristiques d'une observation afin de l'affecter à une classe d'un ensemble prédéfini. Les classes sont discrètes,
- L'estimation : permet d'obtenir une variable continue en combinant les données en entrée. L'estimation est souvent utilisée pour effectuer une tâche de classification en utilisant un barème,
- La prédiction : ressemble à la classification et à l'estimation, mais les enregistrements sont classés selon un certain comportement futur prédit ou à une valeur future estimée s'appuyant sur le passé et le présent, mais le résultat se situe dans un futur généralement précisé,

- La segmentation (classification automatique) : consiste à segmenter une population hétérogène en sous-populations homogènes. Contrairement à la classification, les sous populations ne sont pas préétablises,
- La description : il s'agit de décrire les données d'une base complexe. Cette tâche engendre souvent une exploitation supplémentaire en vue de fournir des explications.

Une fois les tâches identifiées, elles sont utilisées pour restreindre la gamme des méthodes prises en compte. En termes généraux, le but est de sélectionner la technique de fouille de données qui minimise le nombre et la difficulté des transformations de données qui doivent être effectuées pour produire de bons résultats. Les données brutes peuvent demander différentes manières d'être résumées, les valeurs manquantes doivent être traitées, les données redondantes ou non pertinentes doivent être éliminées et les outliers doivent être détectés. Ces transformations sont nécessairement indépendantes de la technique choisie.

1.3 La classification automatique non supervisée : clustering

La classification est une étape importante pour l'analyse de données. Elle consiste à regrouper les observations d'un ensemble de données en classes homogènes. Il existe deux types d'approhes : la classification supervisée et la classification non supervisée, et entre les deux, il existe la classification semi supervisée. La classification supervisée et la classification non supervisée se différencient par leurs méthodes et par leur but [Jain 1999]. L'objectif d'une méthode de classification déborde du cadre strictement exploratoire. C'est la recherche d'une typologie, ou segmentation, c'est-à-dire d'une partition, ou répartition des individus en classes, ou catégories [Saporta 2006]. Ceci est fait en optimisant un critère visant à regrouper les individus dans des classes, chacune le plus homogène possible et, entre elles, le plus distinctes possible. Cet objectif est à distinguer des procédures de discrimination, ou encore de classement, pour lesquelles une typologie est, a priori, connue, au moins pour un échantillon d'apprentissage. Nous sommes dans une situation d'apprentissage non supervisé, ou en anglais de clustering. La classification supervisée est basée sur un ensemble d'observations $\mathcal{A}$ (appelé "ensemble d'apprentissage") de classes connues, le but étant de découvrir la structure des classes à partir de l'ensemble $\mathcal{A}$ afin de pouvoir généraliser cette structure sur un ensemble de données plus large.

L'objectif de la classification supervisée est d'apprendre, à l'aide d'un ensemble d'entraînement, une procédure de classification qui permet de prédire

l'appartenance d'une nouvelle observation à une classe. En d'autres termes, l'objectif est d'identifier les classes auxquelles appartiennent des observations à partir de leurs variables descriptives.

La classification non supervisée (en anglais clustering) consiste à diviser un ensemble de données $\mathcal{A}$ en sous-ensembles, appelés "classes" (en anglais clusters), tel que les observations d'une classe sont similaires et que les observations de classes différentes sont distinctes, afin d'en comprendre la structure sous-jacente [Berkhin 2006]. Par apprentissage, on entend la capacité de généraliser et de résoudre de nouveaux cas à partir des connaissances mémorisées et des expériences réussies dans le passé. Appelé souvent "la branche connexionniste de l'intelligence artificielle", l'apprentissage non supervisé puisait initialement ses sources dans les neurosciences. Au cours des dernières années, il s'est détaché de ses origines pour faire appel à des théories et outils d'autres disciplines : théorie de l'information, traitement du signal, programmation mathématique, statistique, etc.

Des préoccupations convergentes en analyse de données ont donné naissance à la théorie de l'apprentissage statistique [Vapnik 1995]. Il existe trois principales tâches d'apprentissage automatique : apprentissage supervisé, apprentissage non supervisé et apprentissage par renforcement. Pour un problème de classification, un système d'apprentissage supervisé permet de construire une fonction de prise de décision (un classifieur) à partir des actions déjà classées (ensemble d'apprentissage), pour classer des nouvelles actions. Dans le cas de l'apprentissage non supervisé, on dispose d'un nombre fini de données d'apprentissage sans aucune étiquette. L'apprentissage par renforcement a la particularité que les décisions prises par l'algorithme d'apprentissage influent sur l'environnement et les observations futures [Cao 2012]. Nous ne nous intéresserons dans ce travail qu'à la classification non supervisée. Toutes les approches dévelloppées dans cette thèse rentrent dans le cadre totalement non supervisée à l'exception de l'approche GOF-Novelty, que nous développons dans le chapitre 3.

Les méthodes de classification non supervisée ou automatique regroupent les observations en un nombre restreint de classes homogènes et séparées. Homogènes signifie que les observations d'une même classe sont le plus proches possible les unes des autres. Séparés signifie qu'il y a un maximum d'écart entre les classes. La proximité et l'écart ne sont pas nécessairement au sens de distance. L'homogénéité et la séparation rentrent dans le cadre des principes de cohésion et d'isolation de Cormack [Cormack 1971]. Les méthodes de classification automatique déterminent leurs classes à l'aide d'algorithmes formalisés. On parle aussi de méthodes exploratoires, qui ne sont pas explicatives. Les méthodes de classification automatique ont apporté une aide précieuse, notamment par leurs applications en biologie, en médecine, en astronomie et

en chimie. Cormack [Cormack 1971] distingue trois familles de méthodes : la classification hiérarchique, le partitionnement et le groupement. Devant un problème défini de façon aussi imparfaite, il était naturel de voir apparaître un grand nombre de techniques. Récemment, plusieurs autres catégories de méthodes sont rajoutées à la taxonomie de Cormack : la classification automatique sous contraintes [Basu 2008], la classification automatique floue [Bouchon-Meunier 2010, Levillain 2010, Kundu 2012, Kaur 2013] et les méthodes géométriques [Fiori 2012].

1.3.1 Classification par partitionnement

Il existe de nombreuses techniques de classification par partitionnement, la plus connue étant K-means (ou "K-moyennes") [Hartigan 1979] et ses variantes. Un type très particulier de Réseaux de Neurones, les cartes de Kohonen (ou Self Organizing Maps "SOM") [Kohonen 1995] peut être perçu comme une technique de partitionnement puisque cherchant à donner, dans la mesure du possible, une représentation plane des classes qui respecte leurs positionnements relatifs dans l'espace des données.

Il existe deux types de classifications par partitionnement : le partitionnement "doux" et le partitionnement "dur". L'essentiel du partitionnement doux est que chacune des classes réelles, sous-jacentes, occupe une région limitée de l'espace. En particulier, l'Analyse Discriminante nous a habitué à penser en termes de classes ayant des distributions multi-normales, et donc se chevauchant nécessairement. Il est donc naturel de considérer la possibilité que les classes empiètent les unes sur les autres [Cleuziou 2004, Cleuziou 2008, Cleuziou 2013]. Cependant, l'idée générale du partitionnement dur est de découper l'espace des observations en un certain nombre de régions disjointes, définies par des frontières, et de décréter que toutes les observations situées dans une même région de l'espace appartiennent à une même classe. Chaque classe est représentée par un "prototype", observation virtuelle censée être la plus représentative de la population de la classe. Le prototype d'une classe sera le plus souvent le barycentre des observations de la classe. Ces prototypes sont positionnés de façon itérative dans les zones à forte densité, et les observations sont affectées aux classes sur la base d'un critère de proximité aux différents prototypes.

1.3.1.1 K-means

La méthode des K-means [Hartigan 1979] (aussi appelée K-moyennes) est sans doute la méthode la plus utilisée dans cette famille d'approches de classification par partitionnement. Il existe plusieurs extensions de K-means. Nous

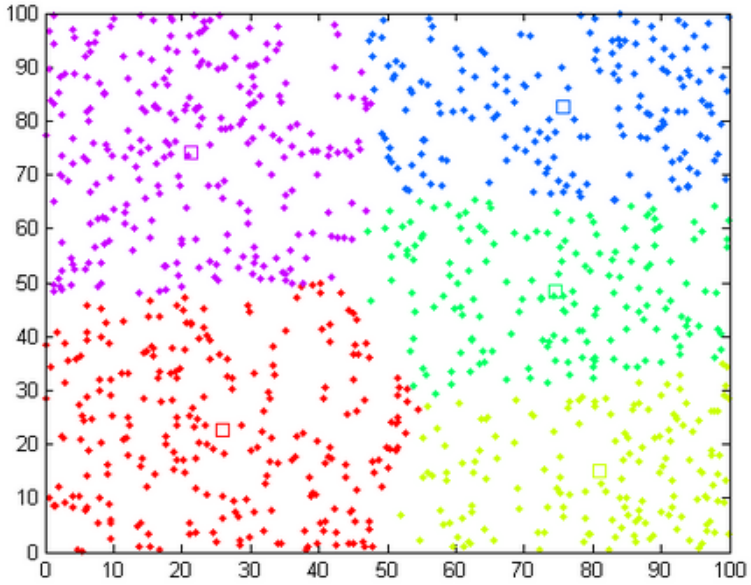


FIGURE 1.2 – Exemple d'une classification par partitionnement.

citons la méthode des centres mobiles [Forgy 1965], la méthodes des nuées dynamiques [Diday 1980], le fuzzy C-means [Pal 2005], le double K-means pour le bi-partitionnement [Govaert 1983, Vichi 2001] et les travaux de Cleuziou [Cleuziou 2008] sur l'extension des K-means pour la recherche de classes recouvrantes. Soit $\mathcal{A} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ l'ensemble des observations d'une base de données, et soit $\mathcal{W} = \{\mathbf{w}_k, \mathbf{w}_k \in \Re^d\}_{c=1}^{K}$ l'ensemble des référents (représentants, prototypes) de chaque classe. Chaque référent est associé à un sous-ensemble de données affectées à la partition c qui est noté P_c. K-means construit K classes à partir d'un ensemble de n individus, tout en minimisant la fonction de coût suivante :

$$\mathcal{J}_{K-means}(\mathcal{W}, \phi) = \sum_{i=1}^{n} \sum_{k=1}^{K} \|\mathbf{x}_i - \mathbf{w}_k\|^2 \tag{1.1}$$

La méthode utilisée pour la minimisation de la fonction $\mathcal{J}_{K-means}(\mathcal{W}, \phi)$ est une méthode itérative dont l'itération de base comporte deux phases :

– **Phase d'affectation :** il s'agit, dans cette phase, de minimiser la fonction $\mathcal{J}_{K-means}(\mathcal{W}, \phi)$ par rapport à la fonction d'affectation ϕ, et de supposer que les vecteurs référents $\mathcal{W}$ sont fixés à la valeur courante ; la minimisation s'obtient en affectant chaque observation $\mathbf{x}$ au référent $\mathbf{w}_c$ à l'aide de la fonction d'affectation ϕ :

$$\phi(\mathbf{x}_i) = \arg \min_{1 \leq k \leq K} \|\mathbf{x}_i - \mathbf{w}_k\|^2 \tag{1.2}$$

La nouvelle fonction d'affectation définit une nouvelle partition $\mathcal{P}$ de l'ensemble $\mathcal{D}$ qui est formée par les référents $\mathbf{w}_c$.

– **Phase de minimisation :** la deuxième phase de l'itération fait décroître à nouveau $\mathcal{J}_{K-means}(\mathcal{W}, \phi)$ en fonction de l'ensemble des référents $\mathcal{W}$. On suppose, dans ce cas, que ϕ est fixé à la valeur courante. Les référents $\mathbf{w}_c$ sont calculés à l'aide de la formule suivante :

$$\mathbf{w}_c = \frac{\sum_{\mathbf{x}_i \in P_c} \mathbf{x}_i}{|P_c|} \tag{1.3}$$

D'un point de vue algorithmique, la méthode des K-means peut se résumer de la manière suivante :

Algorithme 1 : algorithme K-means

1: **Initialisation :**
$t = 0$, choisir un système des référents initial $\mathcal{W}^0$ et le nombre d'itérations N_{iter}

2: **Étape itérative :**
À l'itération t, on suppose connu $\mathcal{W}^{t-1}$ et la fonction d'affectation ϕ^{t-1} calculée à l'itération $t-1$
– **Phase d'affectation :**
Prendre comme nouvelle fonction d'affectation ϕ^t celle qui minimise $\mathcal{J}_{K-means}(\mathcal{W}, \phi)$ par rapport à ϕ et pour $\mathcal{W}^{t-1}$ fixé, cette fonction est définie par l'expression 1.2.
– **Phase de minimisation :**
La fonction d'affectation ϕ^t étant fixée, choisir le système de référents qui minimise la fonction $\mathcal{J}_{K-means}(\mathcal{W}, \phi)$. D'après ce qui précède, chaque référent $\mathbf{w}_r^t$ est calculé selon l'expression 1.3.

3: **Répéter** l'étape itérative jusqu'à atteindre N_{iter} ou une stabilisation de $\mathcal{J}_{K-means}(\mathcal{W}, \phi)$.

L'algorithme des K-means décroît la formule $\mathcal{J}_{K-means}(\mathcal{W}, \phi)$ à chaque itération et converge en un nombre fini d'itérations vers un minimum local de la fonction de coût $\mathcal{J}_{K-means}(\mathcal{W}, \phi)$.

Nombreux sont les avantages des K-means. C'est d'abord un algorithme simple, et compréhensible, sa complexité de calcul en $O(k \times n)$ et enfin applicables à des données de grande taille. Néanmoins, l'algorithme de K-means

présente quelques inconvénients tels que : le nombre des classes doit être fixé au départ, il ne détecte pas les données bruitées, le résultat dépend du tirage initial des centres des classes et la convergence globale de l'algorithme n'est pas garantie.

1.3.2 Classification hiérarchique

La classification hiérarchique [1] est une famille de techniques qui génèrent des suites de partitions emboîtées les unes dans les autres, et allant depuis la partition triviale où chaque observation est une classe à une seule classe contenant toutes les observations. Entre ces deux extrêmes figurent de nombreuses partitions plus réalistes entre lesquelles l'analyste devra choisir.

- les méthodes ascendantes (algorithmes agglomératifs),
- les méthodes descendantes (algorithmes divisifs).

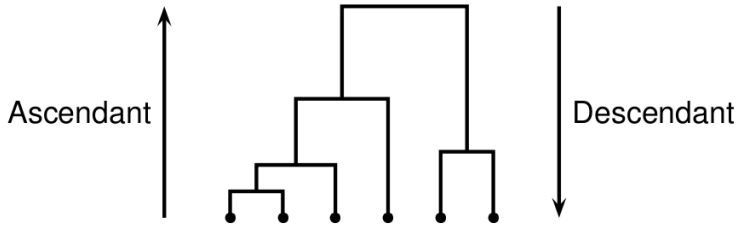


FIGURE 1.3 – Classification ascendante et classification descendante.

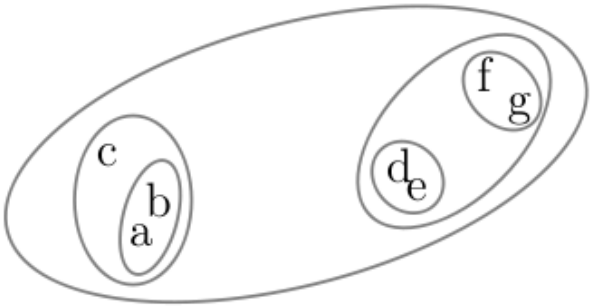


FIGURE 1.4 – Exemple de partitions emboîtées.

1. Voir [Silla 2011] pour plus de précisions sur les différentes approches de classification hiérarchique

1.3.2.1 La Classification Ascendante Hiérarchique (CAH)

Ces méthodes sont les plus anciennes et les plus utilisées dans la classification automatique [Vijaya 2006]. Supposons que nous avons des observations à classer. Les algorithmes agglomératifs définissent d'abord une partition initiale en classes unitaires. Par la suite, ils fusionnent successivement les classes jusqu'à ce que toutes les entités soient dans la même classe. Dans chaque étape de fusion des classes, le décompte des dissimilarités entre les nouvelles classes est nécessaire. Le choix des classes se fait selon le critère qui caractérise la méthode. Les méthodes de cette catégorie diffèrent selon le critère local choisi et selon la méthode de calcul des dissimilarités inter-classes. Nous retrouvons notamment les méthodes issues de la théorie des graphes et les méthodes qui se basent sur la minimisation des carrés des erreurs.

Le but de la classification ascendante hiérarchique est d'obtenir une classification automatique de l'ensemble d'individus. Elle commence par déterminer, parmi les n individus, quels sont les 2 individus qui se ressemblent le plus par rapport à l'ensemble des d variables spécifiées. Elle regroupe alors ces 2 individus pour former une classe. Il existe donc à ce niveau $(n-1)$ classes, une classe est formée des 2 individus regroupés précédemment, les autres ne contenant qu'un unique individu. Le processus se poursuit en déterminant quelles sont les 2 classes qui se ressemblent le plus pour les regrouper. Cette opération est répétée jusqu'à l'obtention d'une unique classe regroupant l'ensemble des individus. Cette procédure est basée sur 2 choix :

- La détermination d'un critère de ressemblance entre les individus.
- La détermination d'une dissimilarité entre les classes : procédé appelé "critère d'agrégation".

Les critères d'agrégation : de nombreux critères d'agrégation ont été proposés dans la littérature, les plus connus sont :

- **Le critère du saut minimal**
  La distance entre 2 classes C_1 et C_2 est définie par la plus courte distance séparant un individu de C_1 et un individu de C_2.

$$D(C_1, C_2) = min(\{d(\mathbf{x}, \mathbf{y})\}, \mathbf{x} \in C_1, \mathbf{y} \in C_2)$$

- **Le critère du saut maximal**
  La distance entre 2 classes C_1 et C_2 est définie par la plus grande distance séparant un individu de C_1 et un individu de C_2.

$$D(C_1, C_2) = max(\{d(\mathbf{x}, \mathbf{y})\}, \mathbf{x} \in C_1, \mathbf{y} \in C_2)$$

- **Le critère de la moyenne**
  Ce critère consiste a calculer la distance moyenne entre tous les éléments

de C_1 et tous les éléments de C_2.

$$D(C_1, C_2) = \frac{1}{n_{C_1} n_{C_2}} \sum_{\mathbf{x} \in C_1} \sum_{\mathbf{y} \in C_2} d(\mathbf{x}, \mathbf{y})$$

Avec :
- n_{C_1} : le cardinal de C_1
- n_{C_2} : le cardinal de C_2

– **Le critère de Ward**
Ce critère ne s'applique que si on est muni d'un espace euclidien. La dissimilarité entre 2 individus doit être égale à la moitié du carré de la distance euclidienne d. Le critère de Ward consiste à choisir, à chaque étape, le regroupement de classes tel que l'augmentation de l'inertie intra-classe soit minimale.

$$D(C_1, C_2) = \frac{n_{C_1} n_{C_2}}{n_{C_1} + n_{C_2}} d(g_{C_1}, g_{C_2})$$

Avec :
- g_{C_1} : le centre de gravité de C_1
- g_{C_2} : le centre de gravité de C_2

– **Le critère des centres de gravité**
La distance entre 2 classes C_1 et C_2 est définie par la distance entre leurs centres de gravité.

$$D(C_1, C_2) = d(g_{C_1}, g_{C_2})$$

La difficulté du choix du critère d'agrégation réside dans le fait que ces critères peuvent déboucher sur des résultats différents. Un des critères les plus couramment utilisés à cet effet est celui du Ward [Zhao 2005].

On retrouve plusieurs variantes de la classification ascendante hiérarchique dans la littérature. CURE [Guha 1998] est un des algorithmes les plus utilisés dans cette famille de méthodes.

Algorithme CURE

Cet algorithme utilise un échantillon représentatif de l'ensemble des données pour réduire la complexité temporelle des calculs. Cet échantillon est divisé en sous-ensembles qui sont regroupés en sous-classes. Les sous-classes sont agrégées hiérarchiquement en utilisant la distance entre deux sous-classes C_1 et C_2. La plus petite distance entre un représentant de C_1 et un représentant de C_2 est calculée. Ce processus est répété jusqu'à l'obtention des K classes demandées. Le détail de cette méthode est décrit dans l'algorithme suivant :

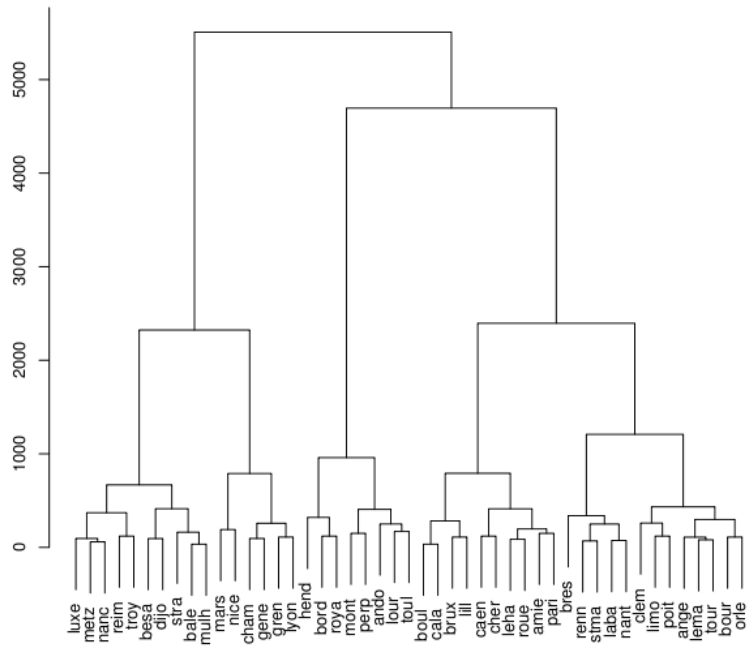


FIGURE 1.5 – Exemple d'un dendrogramme issu de la classification des données par CAH.

Algorithme 2 : Algorithme CURE

1: Entrées : le nombre maximum de classes désiré (K).
2: Extraire un échantillon $A_s (A_s \subset A)$ de taille s de l'ensemble I d'individus.
3: Diviser l'échantillon I_s en p sous-ensembles de taille $\frac{s}{p}$.
4: Déterminer une partition partielle de chaque sous-ensemble en $\frac{s}{pq}$ sous-classes avec $q > 1$.
5: Éliminer les sous-classe d'effectif faible.
6: Déterminer dans chaque sous-classes un ensemble de c points.
7: Agréger de façon hiérarchique les sous-classes $d(C_1, C_2) = min(d(\mathbf{x}, \mathbf{y}))$, $\forall \mathbf{x}$ représentant de C_1 et $\forall \mathbf{y}$ représentant de C_2.
8: Arrêter la procédure d'agrégation quand on obtient K classes.
9: Classer l'ensemble $\mathcal{A}$ total en utilisant les c points représentant les K classes obtenues après l'agrégation. Chaque observation est affectée à la classe possédant le représentant qui lui est le plus proche.

L'algorithme CURE permet de calculer un nombre c constant de données représentatives de chaque cluster. Ces représentants sont calculés itérativement en sélectionnant la donnée la moins similaire au barycentre du cluster,

puis en sélectionnant la donnée la moins similaire à celle venant juste d'être choisie, et ainsi de suite jusqu'à l'obtention de c représentants.

Il existe deux principales limites à l'utilisation de ce type d'algorithme pour l'analyse de données réelles. Premièrement une fois le dendrogramme obtenu, il est nécessaire de choisir un niveau de coupure pour obtenir les clusters. Le choix de ce niveau de coupure reste un problème difficile malgré de nombreuses méthodes proposées dans la littérature [Jain 1999]. Deuxièmement, tous ces algorithmes ont une complexité au minimum proportionnelle au carré du nombre de données, ce qui les rend inutilisables pour l'analyse de grandes base de données.

1.3.2.2 La Classification Descendante Hiérarchique (CDH)

Dans le paragraphe précédent, nous avons vu que la classification ascendante hiérarchique ne se base que sur un seul critère à la fois. Ceci engendre uniquement une séparation (méthode du lien simple) ou une homogénéité (méthode du lien complet) optimale des classes. Ce qui risque de donner naissance à l'effet de chaînage (deux entités très dissimilaires appartenant aux points extrêmes d'une longue chaîne peuvent appartenir à la même classe) ou à l'effet de dissection (deux entités très similaires peuvent être dans deux classes différentes). Pour faire face à ces deux problèmes, nous retrouvons les algorithmes divisifs de la classification descendante hiérarchique. Ces algorithmes commencent par former une seule classe qui englobe toutes les observations. Par la suite, ils choisissent une classe de la partition en cours selon un premier critère local. Ils procèdent ensuite à une bipartition successive selon un deuxième critère local des classes choisies. Cette bipartition continue jusqu'à ce que toutes les entités soient affectées à différentes classes [Murtagh 1983].

À l'inverse de la classification ascendante hiérarchique, à chaque étape de l'algorithme, il y a deux processus à réaliser :

1. Chercher une classe à scinder.
2. Choisir un mode d'affectation des observations aux sous-classes.

Parmi les algorithmes les plus anciens dans cette famille de méthodes figure l'algorithme de Williams et Lambert [Williams 1959].

Algorithme de Williams et Lambert

L'algorithme de Williams et Lambert (1959) [Williams 1959], que nous décrivons ici, est particulièrement rudimentaire. Il n'est applicable que sur des variables qualitatives. En effet, il sélectionne d'abord une variable pour servir de critère d'affectation : tous les individus présentant, pour cette variable, la même modalité sont rangés dans la même classe. La variable retenue est celle

qui, dans la classe C à scinder, est la plus corrélée par rapport à toutes les autres variables. Comme il s'agit de variables qualitatives, alors la corrélation est mesurée par le χ^2 de contingence. Donc les χ de contingence de toutes les variables prises deux à deux sont calculées, et seulement celles pour laquelle la somme de ses χ^2 est maximum sont retenues. La table des χ^2 de contingence entre variables est alors rapide à obtenir, par comparaison au temps qu'il faudrait pour calculer, par exemple, la matrice de *Jaccard* relative aux individus.

1.3.3 Classification par regroupement

Ce type de classification cherche à décomposer directement la base de données dans un ensemble de clusters disjoints. En effet, la classification par regroupement détermine la taille de la partition qui optimise une fonction objectif. La fonction objectif peut mettre en valeur la structure locale ou globale des données. Son optimisation est une procédure itérative.

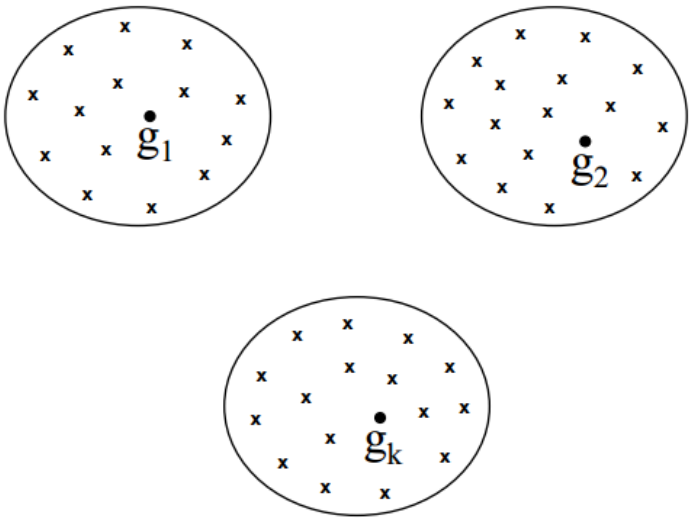


FIGURE 1.6 – Exemple d'une classification par regroupement.

1.3.3.1 Méthodes basées sur la distance et la densité

Les classes rencontrées dans les applications ont souvent des distributions unimodales : à partir d'un noyau central, la densité des observations décroît de façon monotone dans toutes les directions de l'espace. Beaucoup de techniques

de classification non supervisée s'appuient sur cette image et portent une attention particulière aux ensembles d'observations ayant entre elles de faibles distances, c'est-à-dire des régions de forte densité [Kriegel 2011, Nagpal 2011, Munaga 2012, Jahirabadkar 2013]. Elles le font de diverses façons :

- En utilisant les distances entre observations pour construire les classes (par exemple, les méthodes hiérarchiques).
- En reconnaissant que les zones peuplées mais de faible inertie autour de leurs barycentres sont des zones de forte densité (K-means).
- En faisant de l'estimation de densité de façon paramétrique (modèles de mélanges) ou non paramétrique (méthodes basées sur l'estimation de densité par les K plus proches voisins).

L'algorithme DBSCAN est une des méthodes les plus utilisées dans cette famille de modèles. Le principe de ces méthodes est de caractériser une classe comme étant une zone où le nombre de données initiales est plus important qu'ailleurs.

L'algorithme DBSCAN

DBSCAN (Density Based Spatial Clustering of Applications with Noise) [Ester 1996], et ses dérivés tels que OPTICS [Ankerst 1999] ou DBCLASD [Xu 1998], sont basés sur l'idée de définir la notion de voisinage de rayon ε d'un point : tous les points situés à une distance de ce point inférieure à ε appartiennent au voisinage. Pour qu'une classe soit identifiée, il est nécessaire qu'un voisinage contienne plus de "$MinPoints$". Les clusters sont alors agrandis en associant de proche en proche les points de voisinage qui respectent les conditions précédentes.

DBSCAN utilise la notion *connecté-densité* pour former des classes :

- Un point est dit *directement accessible-densité* ($\varepsilon - MinPoints$) d'un autre point s'il se trouve dans le voisinage ($\varepsilon - MinPoints$) de ce point.
- Un point est dit *accessible-densité* ($\varepsilon - MinPoints$) d'un autre point s'il y a une chaîne de points entre eux dont tous les 2 points successifs sont directement *accessibles-densité* ($\varepsilon - MinPoints$).
- Un point est dit *connecté-densité* ($\varepsilon - MinPoints$) d'un autre point s'il y a un point duquel tous les deux points sont *accessibles-densité*.

Une classe avec ε et $MinPoints$ prédéfinis est définie comme un ensemble non vide d'individus qui satisfait 2 conditions :

- La condition de *connectivité*, i.e. tous les points de la classe doivent être *connectés-densité*.
- La condition de *maximum*, i.e. tous les points qui se trouvent dans le

voisinage ($\varepsilon - MinPoints$) d'un point de la classe doivent appartenir à cette classe.

Le bruit est défini comme un ensemble de points qui n'appartiennent à aucune classe. Il y a deux points différents qui sont pris en compte dans la classification :

- Un point *de noyau* : c'est le point qui a un voisinage ($\varepsilon - MinPoints$).
- Un point *non noyau* : c'est celui qui n'a pas un tel voisinage. Un point non noyau peut être un point de frontière ou un bruit.

L'algorithme commence d'un point arbitraire et cherche toutes les observations *accessibles-densité*. S'il est point de noyau, alors cette phase forme une classe. S'il est une observation de frontière et qu'il n'y a aucun point qui est *accessible-densité* à partir de lui, alors c'est un bruit, l'algorithme passe à une autre observation.

Algorithme 3 : Algorithme DBSCAN

1: Entrées : ε, $MinPoints$.
2: Prendre un point $\mathbf{x} \in \mathcal{A}$ aléatoirement.
3: Mettre dans une classe C tous les points *accessibles-densité* à partir de $\mathbf{x}$.
4: Si le point $\mathbf{x}$ est noyau, alors C est une classe.
5: Si $\mathbf{x}$ est un point non noyau (frontière), passer à un autre point et retourner en (3).
6: Répéter les étapes 2, 3, 4 et 5 jusqu'à passer tous les points $\mathbf{x}$.

Cet algorithme présente l'intérêt de trouver lui-même une évaluation du nombre de classes. Celles-ci peuvent avoir des formes arbitraires. L'algorithme permet également de bien gérer les données aberrantes, qui ne sont pas affectées aux clusters détectés. Cependant, DBSCAN requiert des paramètres ε et $MinPoints$, et l'expérience montre que les résultats obtenus sont très sensibles au choix de ces paramètres.

1.3.3.2 Méthodes probabilistes et mélange de modèles

Les données d'une méthode probabiliste sont souvent un échantillon extrait indépendamment d'un modèle de mélanges de plusieurs distributions de probabilités [Babacan 2012]. L'objectif d'une telle méthode est la recherche des centres et des matrices de covariance de ces distributions et les probabilités a priori de façon à maximiser la vraisemblance des données. L'algorithme classique utilisé dans cette famille de méthodes est appellé "EM" (Expectation Maximization). Chaque cluster obtenu est associé aux paramètres de la distribution (moyenne, écart-type, etc.) [Marlin 2012].

Soit un tableau de données $\mathcal{A}$. $\mathcal{A}$ est considéré comme un échantillon $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ i.i.d. de n observations dont la loi de probabilité admet la densité définie par [Jollois 2003] :

$$\forall \mathbf{x}_i \ f(\mathbf{x}_i; \theta) = \sum_{k=1}^{K} \pi_k \varphi_k(\mathbf{x}_i; \alpha_k) \tag{1.4}$$

Avec :

$$\forall k = 1, \ldots, K, \pi_k \in [0, 1] \ \ et \ \ \sum_{k=1}^{K} \pi_k = 1$$

Où :

- $\varphi_k(\mathbf{x}_i; \alpha_k)$ représente la densité de probabilité.
- π_k désigne la probabilité qu'un élément de l'échantillon suive la loi φ.
- $\theta = (\pi_1, \ldots, \pi_K; \alpha_1, \ldots, \alpha_K)$ représente le paramètre inconnu du modèle de mélange.

Le problème revient à estimer $(\pi_1, \ldots, \pi_K)$ et $(\alpha_1, \ldots, \alpha_K)$. Ces paramètres peuvent être estimés par la maximisation de la vraisemblance. La fonction de vraisemblance V s'écrit dans ce cas :

$$\mathcal{V}(\mathbf{x}_1, \ldots, \mathbf{x}_n; \theta) = \prod_{i=1}^{n} f(\mathbf{x}_i; \theta) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \varphi(\mathbf{x}_i; \alpha_k) \tag{1.5}$$

En introduisant la log-vraisemblance, l'équation (1.5) peut être réécrite de la façon suivant :

$$L(\mathbf{x}_1, \ldots, \mathbf{x}_n; \theta) = \sum_{i=1}^{n} log \left(\sum_{k=1}^{K} \pi_k \varphi_k(\mathbf{x}_i; \alpha_k) \right) \tag{1.6}$$

La méthode de maximum de vraisemblance [Wolfe 1970] consiste à maximiser la log-vraisemblance L (voir équation 1.6). Elle permet de résoudre itérativement les équations de vraisemblance, et les algorithmes les plus efficaces sont de type EM (Estimation-Maximisation) de Dempster [Dempster 1977].

L'algorithme EM est un algorithme itératif qui permet de trouver un maximum local de la fonction vraisemblance des observations lorsque chaque observation contient une partie cachée (ou non observée). Ainsi, on suppose que chaque donnée est de type $(\mathbf{x}, \xi)$ où $\mathbf{x}$ est sa partie observable et ξ sa partie cachée non observable. Nous supposons connus d'une manière explicite la forme de la fonction densité jointe $\varphi(\mathbf{x}, \xi; \theta)$ où θ est l'ensemble de paramètres du modèle à estimer. On suppose que l'on dispose d'une série de données indépendantes : $(\mathbf{x}_1, \xi_1), (\mathbf{x}_2, \xi_2), \ldots, (\mathbf{x}_N, \xi_N)$, pour lesquelles $\mathbf{x}_i$ sont les parties

qu'on a réellement observées et les ξ_i sont les parties cachées (donc inconnues).

Nous souhaitons par la suite maximiser le logarithme de la vraisemblance des parties, des données réellement observées $\mathcal{A} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ dont le logarithme est égal à :

$$\ln V(\mathcal{A};\theta) = \ln V(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n;\theta) = \sum_{i=1}^{n} \ln \varphi(\mathbf{x}_i;\theta) \tag{1.7}$$

où $\varphi(\mathbf{x};\theta)$ est la fonction densité de la partie observée $\mathbf{x}$. En pratique $\varphi(\mathbf{x};\theta)$ est calculable en marginalisant la fonction densité $\varphi(\mathbf{x},\xi;\theta)$ ($\varphi(\mathbf{x};\theta) = \int \varphi(\mathbf{x},\xi;\theta)d\xi$), ce qui donne souvent une fonction log-vraisemblance $\ln V(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n;\theta)$ qui n'est pas simple à optimiser.

L'algorithme EM maximise l'expression 1.7 en utilisant le log-vraisemblance des données entières $\ln V(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \xi_1, \ldots, \xi_n;\theta)$. On désigne par la $\mathbf{\Xi} = \{\xi_1, \ldots, \xi_n\}$ l'ensemble des parties correspondantes et non observées. Chaque itération t de l'algorithme EM comporte deux étapes :

- **Etape E (Expectation step)**
  On suppose à cette étape, que la fonction densité de la partie cachée conditionnée par la partie observée $(\xi/\mathbf{x})$ correspond à la valeur du paramètre θ^{t-1} calculée à l'itération précédente (ou égale à l'initialisation θ^0 si $t = 1$) ; cette fonction densité s'écrit donc $(\varphi(\xi/\mathbf{x}, \theta^{t-1}))$. On calcule alors l'espérance :

$$\begin{aligned} Q(\theta, \theta^{t-1}) &= E\left[\ln V(\mathcal{A}, \mathbf{\Xi}/\theta)/\mathcal{A}, \theta^{t-1}\right] \\ &= \int \ln V(\mathcal{A}, \mathbf{\Xi}/\theta)\varphi(\mathbf{\Xi}/\mathcal{A}, \theta^{t-1}) \\ &= \int \ln V(\mathcal{A}, \mathbf{\Xi}/\theta) \prod_{i=1}^{n} \varphi(\xi_i/\mathbf{x}_i, \theta^{t-1})\ d\xi_i \end{aligned} \tag{1.8}$$

  Cette expression qui est parfois appelée "la vraisemblance relative" se justifie "intuitivement". En effet, étant donné qu'on ne connait pas les valeurs des variables cachées ξ_i associées aux observations $\mathbf{x}_i$, on calcule l'espérance du log-vraisemblance relativement aux variables cachées.

- **Etape de maximisation (Maximization step)**
  Ayant calculé $Q(\theta, \theta^{t-1})$ à l'étape E, il s'agit dans cette étape de maximiser cette expression par rapport à θ. On prend alors :

$$\theta' = \arg\max_{\theta} Q(\theta, \theta^{t-1})$$

Il est démontré que chaque itération (EM) fait croître la fonction log-vraisemblance $(\ln V(\mathcal{A}, \theta^t) \geq \ln V(\mathcal{A}, \theta^{t-1}))$ [Dempster 1977]. Ainsi, l'algo-

rithme EM se présente de la manière suivante :

Algorithme 4 : Algorithme EM

1: **Initialisation** : Choisir des paramètres initiaux θ^0 et N_{iter} (le nombre d'itérations) ;
2: **Itération de Base** ($t \geq 1$) :
   – Etape **E** : Estimer l'expression $Q(\theta, \theta^{t-1})$ définie par 1.8.
   – Etape **M** : Maximiser $Q(\theta, \theta^{t-1})$ par rapport à θ, prendre $\theta^t = \arg\max_\theta Q(\theta, \theta^{t-1})$
3: **Répéter** l'itération de base, jusqu'à stabilisation de θ^t ou jusqu'à $t \geq N_{iter}$.

Remarque : l'algorithme EM est largement utilisé en classification pour bâtir de façon itérative, à partir d'un nombre d'observations données, des modèles de mélanges paramétriques.

1.3.4 Classification topologique

Les cartes auto-organisatrices, désignées en anglais par Self-Organizing Maps (SOM), sont proposées par Kohonen [Kohonen 1995]. Elle rentrent dans la catégorie des méthodes de classification par partitionnement. C'est un type de réseau de neurones artificiels qui utilise l'apprentissage non supervisé pour projeter des données multidimensionnelles, sur un espace de faible dimension, souvent en 2D, dans le but d'en faire des tâches de discrétisation, de quantification vectorielle ou de classification. Une carte auto-organisatrice bi-dimensionnelle est composée de cellules disposées sur une grille rectangulaire ou hexagonale. La figure 1.7 représente une carte auto-organisatrice carrée contenant 49 cellules. À chaque cellule est associé un vecteur référent dans l'espace de données (espace d'entrée) et un emplacement sur la carte (espace de sortie) formé par le numéro de ligne et le numéro de colonne de la cellule.

La carte se présente sous forme d'une grille possédant un ordre topologique de K cellules. Les cellules sont réparties sur les nœuds d'un maillage. La prise en compte dans la carte de la notion de proximité impose de définir une relation de voisinage topologique. L'influence mutuelle entre deux cellules c et r est donc définie par la fonction $\mathcal{K}^{\mathcal{T}}(\delta(c,r))$ où $\delta(c,r)$ est la distance de graphe entre les deux cellules c et r. Chaque cellule c de la grille $\mathcal{C}$ est associée à un vecteur référent $\mathbf{w}_c = (\mathbf{w}_c^1, \mathbf{w}_c^2 \ldots, \mathbf{w}_c^j, \ldots, \mathbf{w}_c^d)$ de dimension d. Les référents de la carte sont représentés par $\mathcal{W} = \{\mathbf{w}_c, \mathbf{w}_c \in \Re^d\}_{c=1}^K$. Chaque référent est associé à un sous-ensemble de données affectées à la cellule c, qui est noté P_c. L'ensemble des sous-ensembles forme la partition de l'ensemble des données

$\mathcal{D}$, $\mathcal{P} = \{P_1, \ldots, P_c, \ldots, P_C\}$. La fonction de coût à minimiser est donc :

$$\mathcal{J}_{SOM}(\mathcal{W}, \phi) = \sum_{i=1}^{n} \sum_{c=1}^{K} K^T(\delta(\phi(\mathbf{x_i}), c)) \|\mathbf{w_c} - \mathbf{x_i}\|^2) \tag{1.9}$$

La notion de voisinage est introduite par fonctions noyaux $\mathcal{K}^T(\delta) = e^{\frac{-\delta}{T}}$.
ϕ affecte chaque observation $\mathbf{x}_i$ à une cellule unique de la carte.

Dans les différentes contributions de cette thèse, nous utilisons les

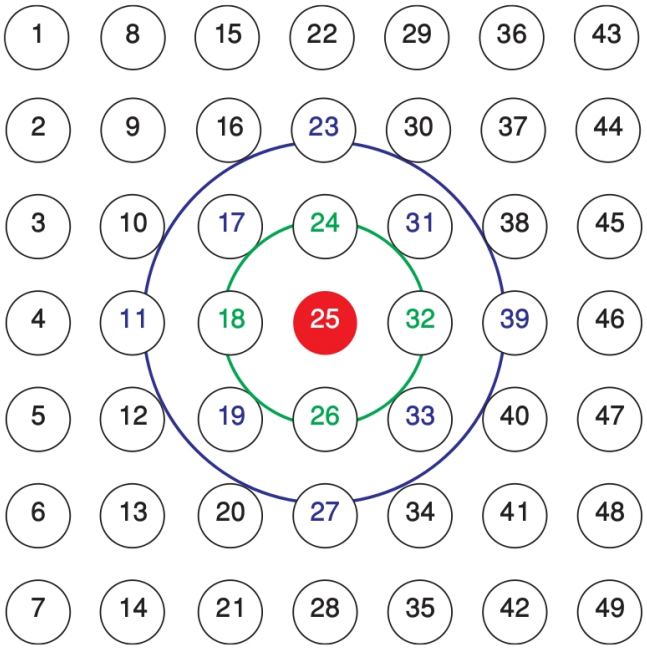


FIGURE 1.7 – Grille carrée de 49 référents.

cartes auto-organisatrices comme modèle d'apprentissage où chaque référent (prototype) représente un cluster/bi-cluster (bloc), tout en préservant la topologie des données. Les individus/variables qui se ressemblent seront affectés au même référent ou à un référent voisin sur la carte.

1.3.4.1 Version stochastique des cartes SOM

Dans la version stochastique, chaque itération consiste à présenter à la carte un vecteur de données choisi au hasard. Le référent dont le vecteur est le plus proche du vecteur d'entrée est appelé "cellule" gagnante ou Best Matching Unit (BMU). Les vecteurs référents de la cellule gagnante et de ses voisins se déplacent vers le vecteur d'entrée (voir figure 1.8). La valeur de ce

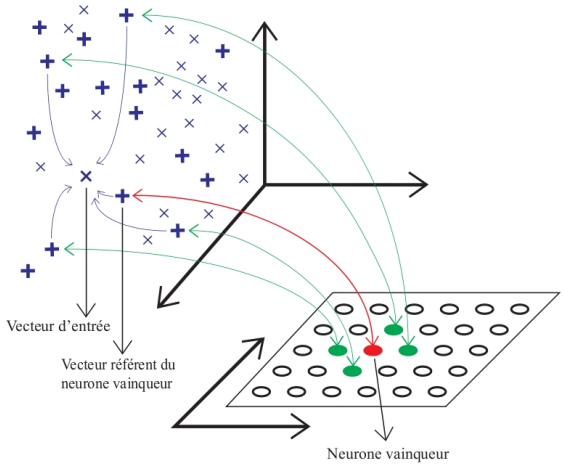


FIGURE 1.8 – Apprentissage de la carte.

déplacement décroît avec le temps et en s'éloignant de la cellule gagnante. Une fonction de voisinage est utilisée pour définir la proximité entre les cellules.

La minimisation de la fonction $\mathcal{J}_{SOM}(\mathcal{W}, \phi)$ (voir formule 1.9), pour une valeur de T fixée, est réalisée par des itérations successives, chacune se décomposant en deux phases. La première phase affecte l'ensemble des observations aux référents, la seconde minimise la valeur de la fonction de coût associée à la partition. L'algorithme stochastique des cartes SOM est composé principalement de 2 phases :

- **Phase d'affectation**
  Il s'agit, dans cette phase, de minimiser la fonction $\mathcal{J}_{SOM}(\mathcal{W}, \phi)$ par rapport à la fonction d'affectation ϕ ; à cette étape, l'ensemble des référents $\mathcal{W}$ est fixé et égal à celui qui est calculé durant la phase précédente. La minimisation s'obtient en affectant chaque observation $\mathbf{x}_i$ au référent $\mathbf{w}_k$ à l'aide de la fonction d'affectation ϕ :

$$\phi(\mathbf{x}_i) = \arg \min_{1 \leq c \leq K} \| \mathbf{x}_i - \mathbf{w}_c \|^2 \tag{1.10}$$

  Cette phase permet de définir une partition de l'ensemble des données $\mathcal{A}$. Chaque observation $\mathbf{x}_i$ étant affectée au référent le plus proche au sens de la distance pondérée.
- **Phase de minimisation**
  La deuxième phase de l'itération fait décroître à nouveau $\mathcal{J}_{SOM}(\mathcal{W}, \phi)$ en fonction de l'ensemble des référents $\mathcal{W}$. Il s'agit maintenant de minimiser $\mathcal{J}_{SOM}(\mathcal{W}, \phi)$ en fonction de l'ensemble des référents $\mathcal{W}$ en supposant ϕ fixée (partition fixée). La fonction $\mathcal{J}_{SOM}(\mathcal{W}, \phi)$ étant convexe par rapport aux paramètres $\mathcal{W}$, la minimisation est obtenue en utilisant la

descente du gradient, l'expression permettant de minimiser la fonction objectif est donnée comme suit :

$$\mathbf{w}_c(t) = \mathbf{w}_c(t-1) - \varepsilon(t)K^T(\delta(\phi(\mathbf{x}_i), c)))\,(\mathbf{w}_c(t-1) - \mathbf{x}_i) \qquad (1.11)$$

Dans la formule 1.11, le pas du gradient $\varepsilon(t)$ décroît au cours des itérations. De la même manière pour le paramètre T. Ceci revient à décroître le voisinage d'influence des référents.

1.3.4.2 Version batch des cartes SOM

Dans la version batch, à chaque itération, tous les vecteurs d'entrée sont présentés au réseau. La cellule vainqueur de chaque vecteur d'entrée est déterminée. Chaque vecteur référent est une moyenne pondérée des vecteurs d'entrée. Les coefficients de pondération sont les valeurs de la fonction de voisinage définissant la proximité entre la cellule vainqueur du vecteur d'entrée et la cellule dont le vecteur référent est calculé.

L'algorithme Batch de Kohonen consiste à utiliser le formalisme des nuées dynamiques pour minimiser la fonction de coût. De la même manière que la version stochastique, le paramètre T décroît au cours des itérations, ce qui revient à décroître le voisinage d'influence des référents. L'algorithme batch des cartes SOM est composé essentiellement de 2 phases :

1. **Phase d'initialisation** :
   Définir la structure et la taille de la carte et les référents initiaux (en général d'une manière aléatoire). Fixer les valeurs de T_{max}, T_{min}. Le nombre d'itérations N_{iter} et prendre la valeur $T = T_{max}$, $t = 0$.
2. **Phase itérative** :
   L'ensemble des référents W^{t-1} de l'étape précédente est connu. Calculer la nouvelle valeur de T en appliquant la formule :

   $$T = T_{max} \times \left(\frac{T_{min}}{T_{max}}\right)^{\frac{t}{N_{iter}-1}}$$

   Effectuer les deux étapes suivantes pour une valeur fixée de T :
   - **Étape d'affectation** : mise à jour de la fonction d'affectation ϕ associée a W^{t-1} . On affecte chaque observation $\mathbf{x_i}$ au référent défini a partir de l'expression suivante :

     $$\phi(\mathbf{x}_i) = \arg\min_{1 \leq c \leq K} \| \mathbf{x}_i - \mathbf{w}_c \|^2 \qquad (1.12)$$

– **Étape de minimisation** : il s'agit maintenant de minimiser $\mathcal{J}_{SOM}(\mathcal{W}, \phi)$ par rapport à l'ensemble des référents $\mathcal{W}$, en supposant que T est fixé à la valeur courante. Pour T fixé, la fonction $\mathcal{J}_{SOM}(\mathcal{W}, \phi)$ est quadratique par rapport à $\mathcal{W}$, elle admet donc un minimum unique qui est atteint pour $\frac{\partial J_{SOM}(\mathcal{W},\phi)}{\partial \mathcal{W}} = 0$. Les nouveaux vecteurs référents sont alors définis par la formule suivante :

$$\mathbf{w}_c^T = \frac{\sum_{r \in \mathcal{C}} K^T(\delta(c,r)) \mathbf{X}_r}{\sum_{r \in \mathcal{C}} K^T(\delta(c,r)) n_r}, \tag{1.13}$$

où $\mathbf{X}_r = \sum_{\substack{\mathbf{x}_i \in \mathcal{A} \\ \phi(\mathbf{x}_i)=r}} \mathbf{x}_i$ représente la somme de toutes les observations de l'ensemble d'apprentissage $\mathcal{A}$ qui ont été affectées à la cellule r. On remarque que chaque référent $\mathbf{w}_c$ ainsi recalculé est le barycentre des vecteurs moyens $\frac{X_r}{n_r}$ des sous-ensembles P_r et que chaque barycentre est pondéré par la valeur $K^T(\delta(c,r))n_r$.

Répéter la phase itérative jusqu'à atteindre $t = N_{iter}$.

1.3.4.3 Ordre topologique des cartes SOM

Dans l'algorithme SOM, le paramètre T joue le rôle d'un paramètre de régulation d'une part, et de la conservation de la topologie d'autre part. Or, dans la pratique, nous souhaitons avoir les deux propriétés : la conservation de la topologie et l'approximation de la fonction densité. C'est pourquoi on fait décroître le paramètre T dans l'intervalle $[T^{max}, T^{min}]$. Si $P_c = \{\mathbf{x}_i, \phi(\mathbf{x}_i) = c\}$, alors la fonction de coût peut être décomposée de la manière suivante :

$$\begin{aligned} J_{SOM}(\mathcal{W}, \phi) &= \left[\sum_c \sum_{r \neq c} \sum_{\mathbf{x}_i \in P_r} K^T(\delta(c,r)) \|\mathbf{x}_i - \mathbf{w}_r\|^2 \right] \\ &+ \left[K^T(\delta(c,c)) \sum_c \sum_{\mathbf{x}_i \in P_c} \|\mathbf{x}_i - \mathbf{w}_c\|^2 \right] \\ = \frac{1}{2} \sum_c \sum_{r \neq c} K^T(\delta(c,r)) &\left[\sum_{\mathbf{x}_i \in P_r} \|\mathbf{x}_i - \mathbf{w}_c\|^2 + \sum_{\mathbf{x}_i \in P_c} \|\mathbf{x}_i - \mathbf{w}_r\|^2 \right] \\ &+ K^T(\delta(c,c)) \sum_c \sum_{\mathbf{x}_i \in P_c} \mathcal{I}_c \end{aligned} \tag{1.14}$$

Avec $\mathcal{I}_c = \sum_{\phi(\mathbf{x}_i)=c} \|\mathbf{x}_i - \mathbf{w}_c\|^2$

La convergence vers la solution peut se décomposer en deux étapes :

- La première étape : correspond aux grandes valeurs de T ; elle a tendance à assurer la conservation de l'ordre topologique. En effet, la minimisation du premier terme de la fonction de coût permet de rapprocher les sous ensembles correspondants à deux cellules voisines sur la carte, afin de conserver l'ordre topologique entre les différents clusters. Ainsi, si c et r sont voisins sur la carte, $\delta(c,r)$ est alors petit et dans ce cas $K^T(\delta(c,r))$ est grand ; la minimisation du premier terme aura pour effet de réduire davantage le terme qui le multiplie $\sum_{\mathbf{x}_i \in P_r} \|\mathbf{x}_i - \mathbf{w}_c\|^2 + \sum_{\mathbf{x}_i \in P_c} \|\mathbf{x}_i - \mathbf{w}_r\|^2$.
- La seconde étape : pour les petites valeurs de T, l'algorithme commence à se rapprocher de l'algorithme des K-means et se confond avec ce dernier lorsque T devient très petit et que $\mathcal{K}(\delta(c,r))$ devient néligeable pour deux référents distincts. On peut donc considérer que la première étape initialise la deuxième étape (K-means) par des référents qui ont comme propriété de respecter l'ordre topologique.

On remarque que pour différentes valeurs de température T chacun des termes aura une importance relative dans la minimisation de la fonction de coût. Pour des grandes valeurs de la température T, le premier terme est prépondérant, dans ce cas, la priorité est donnée à la conservation de la topologie. Plus T est petit, plus le second terme est pris en considération. Ainsi, la priorité est donnée à la détermination des référents représentant des sous-ensembles compacts. Dans ce cas, il s'agit exactement de l'algorithme des K-means. Il est donc possible de dire que les cartes auto-organisatrices permettent d'obtenir une solution régularisée de l'algorithme de K-means : la régularisation étant obtenue grâce à la contrainte d'ordre sur les indices.

Les avantages des cartes topologiques sont nombreux. En effet, elles permettent une classification raffinée des données. Ceci est la conséquence de la prise en compte du voisinage, qui permet une meilleure séparation des sous-ensembles d'observations. D'autres avantages sont la simplicité, la rapidité, le faible temps de calcul de cet algorithme (complexité de calcul est $O(nK)$ pour une itération t) et une visualisation directe des résultats, qui permettent une interprétation rapide des résultats obtenus.

Toutefois, il existe d'importants inconvénients des cartes topologiques. Le nombre maximum de classe étant fixé a priori, ce qui constitue un des grands inconvénients de cet algorithme. L'autre inconvénient est celui de l'extrême sensibilité au choix des conditions initiales. Enfin, les cartes topologiques donnent la plupart du temps une partition localement optimale. Ce désavantage est hérité des algorithmes de K-means, car l'approche des cartes topologiques peut être vue comme une extension des K-means en prenant en

compte le voisinage des données.

1.3.5 Décomposition matricielle pour le clustering

Les méthodes de décomposition matricielle ont été d'abord proposées dans la littérature pour offrir des solutions dans le cadre de l'algèbre linéaire. La formulation initiale est appelée "décomposition LU" qui permet de décomposer une matrice comme produit d'une matrice triangulaire inférieure L (comme lower, inférieure) et une matrice triangulaire supérieure U (comme upper, supérieure). Cette formulation a été largement reprise dans le domaine d'ap-

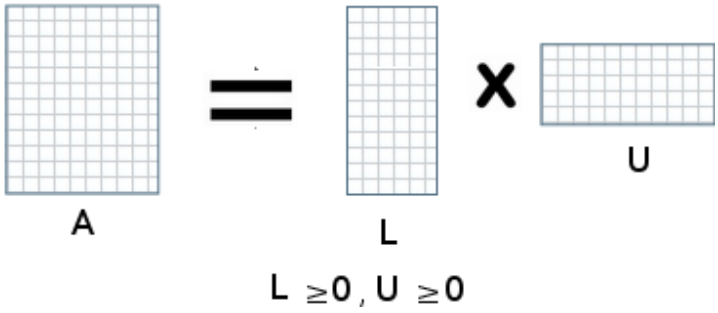


FIGURE 1.9 – Décomposition matricielle pour le clustering.

prentissage non supervisé. Étant donné une matrice $\mathcal{A}$ de dimension $N \times d$ à coefficients non négatifs (positifs ou nuls), la factorisation en matrices non négatives, dont nous utiliserons l'acronyme anglais NMF (Nonnegative Matrix Factorization), consiste à trouver une approximation

$$\mathcal{A} \approx ZW \tag{1.15}$$

telle que les matrices Z et W soient à coefficients non négatifs et de dimensions $N \times K$ et $K \times d$, respectivement. Le "rang" K de la factorisation est souvent choisi tel que $NK+Kd \ll Nd$, produisant une réduction de dimension. Depuis son apparition dans un article de la revue Nature en 1999 [Lee 1999], NMF connaît une forte popularité dans les domaines de l'apprentissage non supervisé. La factorisation est généralement obtenue par résolution du problème de minimisation suivant :

$$\{Z^*, W^*\} = \arg\min_{Z,W} \sum_{i=1}^{N} \sum_{j=1}^{d} (x_{ij} - (ZW)_{ij})^2 = \arg\min_{Z,W} \|X - ZW\|_{Fro}^2 \tag{1.16}$$

Sous contrainte $Z \geq 0$, $W \geq 0$, où la notation $\mathcal{A} \geq 0$ exprime la non négativité des coefficients de $\mathcal{A}$ et non celle des valeurs propres. L'indice Fro désigne

la norme matricielle de Frobenius [Wang 2007]. Différentes implémentations itératives de ce critère sont possibles. Les plus connues sont les méthodes multiplicatives de mise à jour. Nous nous intéressons à l'approche NMF (Non-negative Matrix Factorization) proposée par [Lee 1999]. La mise à jour de Z (resp. W) s'opère avec un coefficient de mise à jour multiplicatif.

Algorithme 5 : Algorithme NMF

1: Entrées :
   – Matrice de données $\mathcal{A}$.
   – Rang de la factorisation k.
2: Sorties :
   – Z et W : matrice d'approximation de la matrice de données initiale $\mathcal{A}$.
3: Initialiser Z et W.
4: $W^{(t+1)} = W^{(t)} \times \frac{(Z^T \mathcal{A})}{(Z^T Z W)}$
5: $Z^{(t+1)} = Z^{(t)} \times \frac{(\mathcal{A} W^T)}{(Z W W^T)}$
6: Répéter les phases 4 et 5 jusqu'à la stabilisation.

Avec les mises à jour de l'algorithme 5, la norme de Frobenius est non croissante, au pire, elle devient invariante si l'on a atteint un point stationnaire. Or, un point stationnaire n'est pas une garantie d'un minimum global du critère. Ceci représente un des inconvénients d'une telle approche.

1.3.6 Autres types de clustering

Dans les mélanges de modèles, les classes se chevauchent, mais chaque observation appartient à une classe et une seule (bien que la détermination de cette classe ne puisse être faite que de façon probabiliste). En classification floue, à l'inverse, on reconnaît qu'une observation peut effectivement appartenir simultanément à plusieurs classes à des degrés divers dont la somme est égale à 1. Cette attitude reflète le fait que, dans de nombreuses situations réelles, les définitions des classes (supervisées ou non) ne rendent pas obligatoire l'appartenance à une classe et une seule [Bouchachia 2013]. La classification floue affecte alors à chaque observation des degrés d'appartenance aux diverses classes d'une façon cohérente avec la répartition géométrique des données. Pour chaque observation, la somme des degrés d'appartenance à chacune des classes est égale à 1 [Bouchachia 2011].

Mentionnons qu'il est possible de procéder à une classification non supervisée non pas sur des observations mais sur des variables. La dissimilarité entre deux variables (similaire à la distance entre observations) est en général définie à partir de leur coefficient de corrélation, des variables fortement corrélées étant alors considérées comme "proches". Une telle classification est peut-être

utile lorsque les variables sont nombreuses et présentent un fort risque de colinéarité. Après la classification, toutes les variables d'une classe seront alors remplacées par une unique variable synthétique représentant au mieux l'ensemble des variables de la classe [Scherrer 2012].

La classification à base de grille est un autre type de méthode de classification. En effet, ces algorithmes sont principalement proposés pour des données spatiales. La caractéristique principale de ce type d'algorithme est qu'il quantifie l'espace des données dans un nombre fini de cellules et qu'il réalise ensuite toutes les opérations dans cet espace.

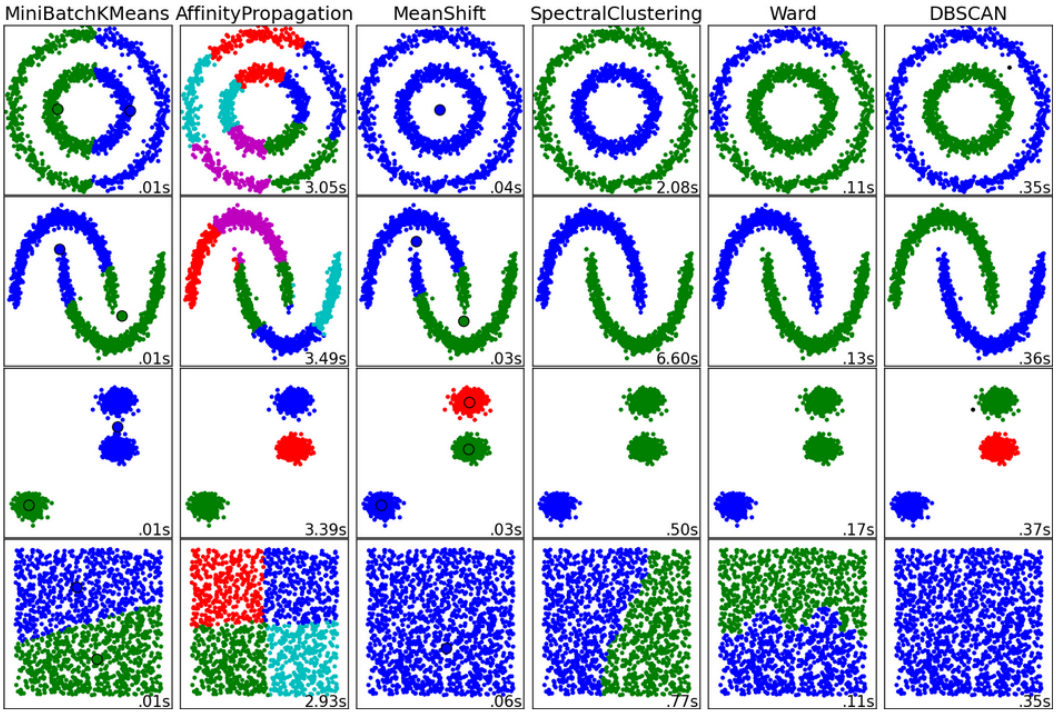


FIGURE 1.10 – Comparaison de quelques méthodes de clustering : K-means, AffinityPropagation, MeanShift, Spectral clustering ; Ward et DBSCAN.

1.4 La classification croisée : bi-clustering

Dans le domaine de la classification, bien que la plupart des méthodes utilisées cherchent à construire des partitions soit sur l'ensemble des observations soit sur celui des variables séparément, il existe d'autres méthodes de classifications croisées qui considèrent simultanément les deux ensembles [Govaert 1983, Abdullah 2006, Govaert 2009, Kwon 2010,

Ayadi 2012, De France 2013]. Comparée à la classification classique, en ne privilégiant pas un ensemble sur un autre, la classification croisée est plus efficace pour découvrir des blocs homogènes dans une matrice de données. Ces dernières années, cette famille d'approches a suscité un grand intérêt dans différentes communautés scientifiques et dans des domaines variés tels que la fouille de données.

Les approches de bi-clustering (bi-partitionnement) sont devenues un sujet d'intérêt majeur en raison de ses nombreuses applications dans le domaine de l'exploration des données. Une méthode de bi-partitionnement, aussi appelée "bi-clustering", co-clustering ou classification croisée, est une méthode d'analyse qui vise à regrouper des données en fonction de leur similarité. La stratégie classique des méthodes de bi-partitionnement cherche à trouver des sous-matrices ou des blocs, qui représentent des sous-groupes de lignes et des sous-groupes de colonnes d'une matrice de données.

Un des objectifs d'une méthode de classification croisée est la recherche d'un couple de partitions, l'une sur les observations (les lignes d'une matrice de données), l'autre sur les colonnes (colonnes d'une matrice de données), tel que la "perte d'information" due au regroupement soit minimale ; c'est-à-dire de sorte que la différence entre l'information apportée par le tableau initial et celle apportée par le tableau obtenu après regroupement soit minimale.

Depuis le premier algorithme de bi-partitionnement, appelé "Block Clustering", proposé par [Hartigan 1972], de nombreuses techniques ont été proposées, telles que l'énumération exhaustive ([Tanay 2002]), l'analyse spectrale ([Greene 2010]), les réseaux bayésiens ([Shan 2010]) et d'autres ([Angiulli 2006]). Les auteurs de [Charrad 2008] classifient les méthodes de bi-partitionnement en quatre grandes familles : la familles des méthodes divisives, la famille des méthodes constructives, la famille des méthodes probabilistes et la famille des méthodes basées sur des algorithmes de partitionnement simple.

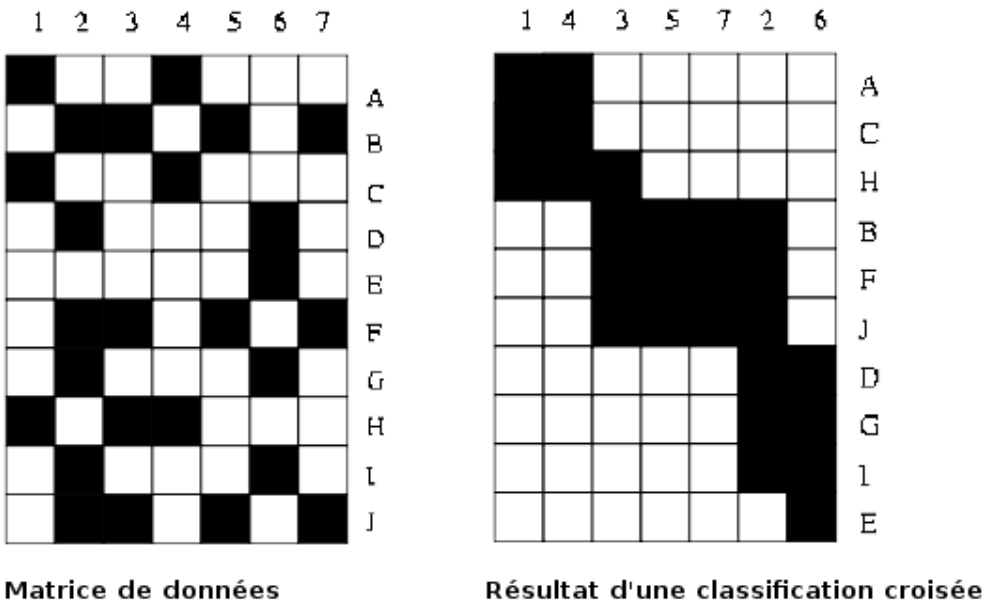


FIGURE 1.11 – Exemple d'une classification croisée d'une marice de données binaire.

1.4.1 Méthodes basées sur des algorithmes de partitionnement simple

Les algorithmes de type K-means ont longuement été utilisés dans le bi-partitionnement. En effet, [Govaert 1983] a défini un algorithme de bi-partitionnement nommé "Croeuc" qui consiste à déterminer une série de couples de partitions minimisant une fonction de coût sur la matrice des données en appliquant le K-means alternativement sur les lignes et les colonnes. L'algorithme Croeuc est proposé pour les données continues. Soit une matrice de données notée $\mathcal{A}$ avec N individus et d variables, x_i^j tels que $1 < i < N, 1 < j < d$ sont les éléments de la matrice $\mathcal{A}$. Les individus sont partitionnés en K classes. De même, les variables sont partitionnées en L classes. P_k et Q_l représentent respectivement les partitions des lignes et des colonnes. Les partitions optimales P et Q sont obtenues grâce à un algorithme itératif qui utilise la somme des distances euclidiennes comme une mesure de l'écart entre la matrice de données $\mathcal{A}$. L'objectif de l'algorithme Croeuc est de trouver une paire de partitions (P, Q) et g, telles que le critère suivant soit

minimisé [Jollois 2003] : [2]

$$W(P,Q,g) = \sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{i\in P_k}\sum_{j\in Q_l}(x_i^j - g_k^l)^2 \tag{1.17}$$

où

- $P = (P_1, \ldots, P_K)$ est la partition des individus en K classes,
- $Q = (Q_1 \ldots, Q_L)$ est la partition des variables en L classes,
- g_k^l est le centre du bloc $\mathbf{x}_k^l$ (prototype).

Il est facile de voir que, pour (P,Q) fixé, les valeurs optimales de g_k^l sont les moyennes de chaque x_i^j appartenant au bloc $\mathbf{x}_k^l$. Les différentes étapes de l'algorithme Croeuc sont :

Algorithme 6 : Algorithme Croeuc

1: Démarrer d'une position initiale (P^0, Q^0, g^0)
2: Calculer $(P^{(c+1)}, Q^{(c+1)}, g^{(c+1)})$ à partir de $(P^{(c)}, Q^{(c)}, g^{(c)})$:
   - 2(a) Calculer $(P^{(c)}, Q^{(c)}, g^{'})$ à partir de $(P^{(c)}, Q^{(c)}, g^{(c)})$,
   - 2(b) Calculer $(P^{(c+1)}, Q^{(c+1)}, g^{(c+1)})$ à partir de $(P^{(c)}, Q^{(c)}, g^{'})$.
3: Recommencer l'étape 2 jusqu'à la convergence de l'algorithme.

Il est à noter que, dans l'étape 2, l'algorithme 6 utilise un double K-means (première phase K-means sur les lignes, deuxième phase K-means sur les colonnes). Ce qui revient, donc, à optimiser alternativement les critères suivants (déduit à partir de 1.17) :

$$W(P, g/Q) = \sum_{k=1}^{K}\sum_{i\in P_k}\sum_{l}^{L}|Q_l|(u_i^l - g_k^l)^2 \tag{1.18}$$

Où $u_i^l = \frac{\sum_{j\in Q_l} x_i^j}{|Q_l|}$, et

$$W(Q, g/P) = \sum_{l=1}^{L}\sum_{j\in Q_l}\sum_{k=1}^{K}|P_k|(v_k^j - g_k^l)^2 \tag{1.19}$$

Où $v_k^j = \frac{\sum_{i\in P_k} x_i^j}{|P_l|}$

L'étape 2(a) de l'algorithme 6 est effectuée par l'algorithme K-means en utilisant la matrice u_i^l. Alternativement, l'étape 2(b) est obtenue par l'algorithme K-means en utilisant cette fois-ci la matrice v_k^j. Ainsi, à la convergence,

2. Nous avons repris dans cette partie de travail les mêmes notations que celle utilisées dans la thèse de Xavier Jollois [Jollois 2003].

des blocs homogènes sont obtenus en réorganisant les lignes et les colonnes selon les partitions P et Q. Chaque bloc (k, l), défini par les éléments x_i^j pour $i \in P_k$ et $j \in Q_l$ est caractérisé par g_k^l.

L'intérêt de cet algorithme a été mis en évidence en comparaison avec K-means appliqué séparément sur les observations et les variables d'une matrice de données [Nadif 2004]. Par sa simplicité et sa rapidité, l'algorithme Croeuc peut s'appliquer sur des données comparables de grande taille. Cependant, il requiert la connaissance du nombre de classes en lignes et en colonnes.

1.4.2 Méthodes probabilistes

Dans la plupart des cas, les méthodes probabilistes sont des méthodes basées sur les modèles de mélanges. Les modèles de mélanges finis de lois de probabilité sont particulièrement utilisés dans le bi-partitionnement. Leur utilisation, comme dans le clustering, revient à supposer que les individus à classifier sont issus d'un modèle de mélange dont chaque composant représente une classe.

Les auteurs affirment que les données d'un modèle de mélange fini de lois de probabilité $\mathcal{A} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ constituent un échantillon de n réalisations indépendantes d'une variable aléatoire dont la fonction de densité peut s'écrire sous la forme de l'équation 1.4 détaillée dans la section 1.3.3.2.

Modèle de mélange pour la classification croisée

Comme indiqué dans l'article [Govaert 2009], la formulation du problème du bi-partitionnement utilise le modèle de mélange classique (équation 1.4), dans lequel la partition des variables $\mathbf{w}$ est considérée comme un paramètre du modèle. La densité du mélange s'écrit de la manière suivante :

$$f(\mathbf{x}_i; \theta) = \sum_k \pi_k \varphi_k(\mathbf{x}_i; \mathbf{w}, \alpha) \tag{1.20}$$

$$\varphi_k(\mathbf{x}_i; \mathbf{w}, \alpha) = \prod_{j,l} \left(\frac{1}{\sqrt{2\pi\sigma_{kl}^2}} \exp^{-\frac{1}{2\sigma_{kl}^2}(\mathbf{x}_{ij} - \mu_{kl})^2} \right)^{w_{jl}} \tag{1.21}$$

- $\theta = (\pi, w, \alpha)$ représente le paramètre du modèle de mélange qui est formé par les proportions $\pi = (\pi_1, \ldots, \pi_g)$,
- la partition des variables $\mathbf{w}$ et les paramètres de chaque composant $\alpha = (\mu_{11}, \ldots, \mu_{gm}, \sigma_{11}^2, \ldots, \sigma_{gm}^2)$, où les μ_k et les σ_k représentent les moyennes et les variances de chaque bloc.

La log-vraisemblance s'écrit comme suit :

$$L(\theta) = log\ f(\mathbf{x}; \theta) = \sum_i log \sum_k \pi_k \varphi_k(\mathbf{x}_i; \mathbf{w}, \alpha) \tag{1.22}$$

Soient : $\mathbf{z}_k = \sum_i \mathbf{z}_{ik}$ et $\mathbf{w}_l = \sum_j \mathbf{w}_{jl}$ les cardinaux de chaque classe, la log-vraisemblance classifiante vérifie :

$$L_c(\mathbf{z}; \mathbf{w}, \theta) = \sum_{i,k} z_{ik} \; log \; (\pi_k \varphi_k(\mathbf{x}_i; \mathbf{w}, \alpha)) \tag{1.23}$$

Dans le cas où la constante additive est égale à $\frac{nd}{2} \; log \; 2 \; \pi$, l'équation 1.23 prend la forme suivante :

$$L_c(\mathbf{z}; \mathbf{w}, \theta) = \sum_k \mathbf{z}_k \; log \; \pi_k - \frac{1}{2} \sum_{i,j,k,l} \mathbf{z}_{ik} \mathbf{w}_{jl} \left(log \; \sigma_{kl}^2 + \frac{1}{\sigma_{kl}^2} (\mathbf{x}_{ij} - \mu_{kl})^2 \right) \tag{1.24}$$

L'écriture de la log-vraisemblance classifiante L_c, définie pour une partition $\mathbf{z}$, peut être, alors, étendue à la partition floue $\mathbf{s} = (\mathbf{s}_i^k; i = 1, \ldots, n; k = 1, \ldots, \mathbf{g})$ associée à la matrice de classification définie par les probabilités conditionnelles [Govaert 2009].

$$L_c(\mathbf{s}; \mathbf{w}, \theta) = \sum_{i,k} s_{ik} \; log \; (\pi_k \varphi_k(\mathbf{x}_i; \mathbf{w}, \alpha)) \tag{1.25}$$

Qui peut s'écrire :

$$L_c(\mathbf{s}; \mathbf{w}, \theta) = \sum_k s_k \; log \; \pi_k - \frac{1}{2} \sum_{i,j,k,l} s_{ik} w_{jl} \left(log \; \sigma_{kl}^2 + \frac{1}{\sigma_{kl}^2} (x_{ij} - \mu_{kl})^2 \right) \tag{1.26}$$

Où $s_k = \sum_i s_{ik}$.

Dans [Govaert 2009], les auteurs ont utilisé l'algorithme EM généralisé (GEM) maximisant la vraisemblance des données observées afin d'estimer les paramètres du modèle. À partir d'une position initiale $(w(0), \theta(0))$, les différentes étapes de cet algorithme EM sont décrites comme suit :

- Étape E

Cette étape se réduit au calcul des probabilités conditionnelles a posteriori, s_{ik}^c

$$s_{ik}^{(c)} = \frac{\pi_k^c \varphi_k(\mathbf{x}_i; \mathbf{w}^{(c)}, \alpha^{(c)})}{\sum_{k'} \pi_{k'}^{(c)} \varphi_{k'}(\mathbf{x}_i; \mathbf{w}^{(c)}, \alpha^{(c)})} \tag{1.27}$$

Ces probabilités conditionnelles peuvent s'écrire $s_{ik} = \frac{e^{s_{ik}}}{\sum_{k'} e^{s_{ik'}}}$ où

$$S_{ik} = log(\pi_k \varphi_k(\mathbf{x}_i; \mathbf{w}, \alpha)) \tag{1.28}$$

Après quelques calculs algébriques, le terme S_{ik} prend la forme suivante :

$$log \pi_k - \frac{1}{2} \sum_l \left(w_l \; log \; \sigma_{kl}^2 + \frac{1}{\sigma_{kl}^2} (e_{il} + w_l (u_{il} - \mu_{kl})^2 \right) \tag{1.29}$$

Avec :
$u_{il} = \frac{\sum_j w_{jl}\ x_{ij}}{w_l}$ et $e_{il} = \sum_j w_{jl}(x_{ij} - u_{il})^2$, plus facile à calculer que la probabilité initiales s_{ik}

- Étape M

Dans cet algorithme, on applique EM généralisé (Generalized EM algorithm, GEM) [Dempster 1977] pour faire croître la quantité $Q(\theta, \theta^{(c)})$. Sachant que l'espérance conditionnelle $Q(\theta, \theta^{(c)})$ peut aussi s'exprimer comme la log-vraisemblance classifiante floue $L_c(s^{(c)}, w, \theta)$, cette fonction Q peut aussi s'écrire :

$$\sum_k s_k^{(c)} log\ \pi_k - \frac{1}{2} \sum_{i,j,k,l} s_{ik}^{(c)} w_{jl} \left(log\ \sigma_{kl}^2 + \frac{1}{\sigma_{kl}^2}(x_{ij} - \mu_{kl})^2 \right) \tag{1.30}$$

Pour faire croître Q, les auteurs proposent d'itérer jusqu'à la convergence des deux étapes suivantes : maximisation de $Q(\theta, \theta^{(c)})$ en $\mathbf{w}$ pour s et $\theta(c)$ fixés puis maximisation de $Q(\theta, \theta^{(c)})$ en θ pour $\mathbf{w}$ et s fixés.
Calcul de $\mathbf{w}$:
Cette étape consiste à maximiser $Q(\theta, \theta^{(c)})$ en $\mathbf{w}$. L'expression 1.30 de $L_c(\mathbf{s}^{(c)}, \mathbf{w}, \theta)$ peut s'écrire :

$$\sum_k s_k^{(c)} log\ \pi_k + \sum_{j,l} w_{jl}\ T_{jl}^{(c)} \tag{1.31}$$

Où : $T_{jl}^{(c)} = -\frac{1}{2} \sum_{i,k} s_{ik}^{(c)} \left(log\ \sigma_{kl}^2 + \frac{1}{\sigma_{kl}^2}(x_{ij} - \mu_{kl}^2) \right)$. La variable j appartient à la classe maximisant $T_{jl}^{(c)}$:

$$w_{jl}^{(c)} = \begin{cases} 1 & \text{si } l = \arg\max_{l'=1,\ldots,m} T_{jl'}^{(c)}; \\ 0 & \text{sinon.} \end{cases}$$

Comme pour le calcul de S_{ik}, les auteurs ont montré que le terme T_{jl} prend la forme suivante :

$$-\frac{1}{2} \sum_k \left(s_k^{(c)} log\ \sigma_{kl}^2 + \frac{1}{\sigma_{kl}^2}(f_{jk} + s_k(v_{kj} - \mu_{kl})^2 \right). \tag{1.32}$$

Où :
$v_{kj} = \frac{\sum_i s_{ik}\ x_{ij}}{s_k}$ et $f_{jk} = \sum_i s_{ik}(x_{ij} - v_{jk})^2$
Calcul de α à partir de $\mathbf{w}$ et $\mathbf{s}$
Cette étape consiste à maximiser $Q(\theta|\theta^{(c)})$ en π et α =

$(\mu_{11}, \ldots, \mu_{gm}, \sigma^2_{11}, \ldots, \sigma^2_{gm})$. En écrivant la log-vraisemblance classifiante sous la forme :

$$L_c(\mathbf{s}, \mathbf{w}, \theta) = \sum_k s_k log\ \pi_k - \frac{1}{2} \sum_{k,l} \left(s_k w_j log\ \sigma^2_{kl} + \frac{1}{\sigma^2_{kl}} \sum_{i,j} s_{ik} w_{jl} (x_{ij} - \mu_{kl})^2 \right) \tag{1.33}$$

Alors : $\pi_k^{(c+1)} = \frac{s_k^{(c)}}{n}$, $\mu_{kl}^{(c+1)} = \frac{\sum_{ij} s_{ik}^{(c)}\ w_{jl}^{(c)} x_{ij}}{s_k^{(c)}\ w_l^{(c)}}$ et $(\sigma^2_{kl})^{(c+1)} = \frac{\sum_{ij} s_{ik}^{(c)}\ w_{jl}^{(c)} (x_{ij} - \mu_{kl})^2}{s_k^{(c)}\ w_l^{(c)}}$

Ces calculs peuvent être optimisés en utilisant les valeurs v_{jk} et f_{jk} définies précédemment, ce qui permet d'accélérer cette étape. Le centre et la variance de chaque bloc sont : $\mu_{kl}^{(c+1)} = \frac{\sum_j w_{jl}^{(c)} v_{jk}}{s_k^{(c)}\ w_l^{(c)}}$ et $(\sigma^2_{kl})^{(c+1)} = \frac{\sum_j w_{jl}^{(c)} \left(f_{jk} + s_k^{(c)} (v_{jk} - \mu_{kl})^2 \right)}{s_k^{(c)}\ w_l^{(c)}}$

1.4.3 Méthodes topologiques

Les méthodes de bi-partitionnement utilisant les cartes auto-organisatrices (SOM) ([Kohonen 2001]) ont été définies par plusieurs auteurs (DCC [Busygin 2002], KDISJ [Cottrell 2004], BCDSM [Benabdeslem 2012], etc.). Ce type de méthodes, rentre dans la catégorie des approches basées sur le partitionnement car, souvent, elles utilisent des algorithmes de classification simple appliqués séparément sur les lignes et les colonnes d'une matrice des données.

Stanislav et al. [Busygin 2002] ont proposé l'approche DCC (Double Conjugated Clustering), qui permet de partitionner l'ensemble des lignes et l'ensemble des colonnes à l'aide des cartes auto-organisatrices. Le principe de base de cette approche et celui de relier les deux partitions par l'intermédiaire d'une bijection associant à chaque référent de l'un des deux espaces un référent de l'autre espace appelé "conjugué". Cette méthode présente l'avantage de convergence relativement rapide et aboutit à la construction de deux partitions, une dans l'espace des lignes et l'autre dans l'espace des colonnes. Chacune de ces partitions est le conjugué de l'autre.

Un des algorithmes que nous retrouvons fréquemment dans la littérature est celui introduit par Corttell, appellé KDISJ [Cottrell 2004]. KDISJ est une variante des cartes topologiques pour le traitement des variables qualitatives d'un tableau de données.

Algorithme KDISJ

Nous rappelons qu'un tableau disjonctif complet consiste à coder des variables qualitatives avec le code 1 pour la modalité observée et 0 pour toutes les autres modalités. Le codage disjonctif complet, permet donc, de transformer des variables qualitatives en des variables de type quantitatif entre lesquelles il est permis de calculer des corrélations.

Expérimentation	Expérimentateur
Test 1	Expérimentateur 1
Test 2	Expérimentateur 2
Test 3	Expérimentateur 3
Test 4	Expérimentateur 1

TABLE 1.1 – Exemple d'un tableau contenant des variables qualitatives.

Expérimentation	Expérimentateur 1	Expérimentateur 2	Expérimentateur 3
Test 1	1	0	0
Test 2	0	1	0
Test 3	0	0	1
Test 4	1	0	0

TABLE 1.2 – Exemple d'un tableau disjonctif complet.

KDISJ (Kohonen for Disjonctive Table) [Cottrell 2004] permet de classer simultanément les observations et les variables qualitatives qui les décrivent. Soit une matrice de données $\mathcal{A}$. Soit d_{ij} le terme général de cette matrice qui peut être considérée comme un tableau de contingence croisant la variable "individu" à N modalités, et la variable "modalités" à M modalités. Le terme d_{ij} prend ses valeurs dans $\{0, 1\}$. La distance χ^2 est utilisée sur les lignes et sur les colonnes. Ensuite, les modalités sont pondérées pour corriger le tableau disjonctif complet de la façon suivante :

$$d_{ij}^c = \frac{d_{ij}}{\sqrt{d_i.d_j}} \tag{1.34}$$

Où : $d_i = \sum_{j=1}^{M} d_{ij}$ et $d_j = \sum_{i=1}^{N} d_{ij}$

Dans le cas d'un tableau disjonctif complet, d_i vaut k, quel que soit i. Le terme d_j est l'effectif de la modalité j. Le tableau ainsi corrigé est noté

D^c (tableau disjonctif corrigé). Après cette transformation, il est possible d'utiliser la distance euclidienne sur D^c qui est équivalente à χ^2 pondérée sur D. Ces corrections sont équivalentes à celles utilisées traditionnellement dans l'analyse des correspondances, qui revient en fait à une analyse en composantes principales pondérée, utilisant la distance χ^2 simultanée sur les lignes et les colonnes. Le passage aux cartes topologiques se fait en utilisant l'architecture classique du modèle SOM en associant à chaque référent **w** un vecteur référent C_w formé de $(M + N)$ composantes, les M premières évoluent dans l'espace des individus (représentés par les lignes de D^c), les N dernières dans l'espace des modalités (représentées par les colonnes de D^c).

La notation :

$$C_w = (C_M + C_N)_w = (C_{M,w} + C_{N,x}) \tag{1.35}$$

permet de mettre en évidence la structure du vecteur référent C_w. Les étapes d'apprentissage de la carte topologique sont doubles. Une ligne de D^c (c'est-à-dire un individu i), puis une colonne (c'est-à-dire une modalité j) sont tirées alternativement. Quand un individu i est tiré, la modalité $j(i)$ lui est associée. Elle est définie par :

$$j(i) = \arg\max_j \ d^c_{ij} \tag{1.36}$$

qui maximise le coefficient d^c_{ij}, c'est-à-dire la modalité la plus rare dans la population totale parmi les modalités qui lui correspondent. Ensuite, un vecteur individu étendu

$$X = (i, j(i)) = (XM, XN)$$

de dimension $(M + N)$ est créé. Ensuite, la procedure cherche parmi les vecteurs-codes celui qui est le plus proche, au sens de la distance euclidienne restreinte aux M premières composantes.

Soit w_0 le référent gagnant. L'étape de minimisation est formulée comme suit :

$$\begin{cases} w_0 = \arg\min_w \ \|X_M - C_{M,w}\| \\ C_w^{(t)} = C_w^{(t-1)} + \varepsilon \ \mathcal{K}(w, w_0)(X - C_w^{(t-1)}) \ . \end{cases}$$

Où ε est le pas d'apprentissage et $\mathcal{K}$ est le rayon de voisinage de la carte.

Quand une modalité j de dimension N (une colonne de D^c) est tirée, l'algorithme de [Cottrell 2004] cherche parmi les vecteurs-codes celui qui est le plus proche, au sens de la distance euclidienne restreinte aux N dernières composantes. Soit z_0 l'unité gagnante. La procédure rapproche les N dernières composantes du vecteur-code gagnant associé à z_0 et de ses voisins de celles du vecteur modalité j, sans modifier les M premières composantes. Soit Y le vecteur colonne de dimension N correspondant à la modalité j. Cette étape

peut s'écrire :

$$\begin{cases} z_0 = \arg\min_{w} \ \|Y - C_{N,w}\| \\ C_{N,u}^{(t)} = C_{N,u}^{(t-1)} + \varepsilon \ \mathcal{K}(w, w_0)(Y - C_{N,u}^{(t-1)}) \ . \end{cases}$$

Après convergence, les individus et les modalités sont classés dans les classes de la carte obtenue. Les individus ou modalités "proches" sont classés dans la même classe ou dans des classes voisines. On appelle KDISJ l'algorithme ainsi défini.

1.4.4 Méthodes divisives

La stratégie de base de ce type de méthode est le découpage itératif de la base de données, qui permet de retrouver les blocs de données qui optimisent certains critères. En effet, au lieu de proposer seulement une partition en lignes et une partition en colonnes, ce type de méthode propose un découpage en blocs homogènes des données. Un des algorithmes les plus anciens et les plus utilisés est One-way Splitting [Hartigan 1972]. Il fait partie des algorithmes dits "divisifs" et permet de diviser une matrice de données en plusieurs sous-matrices correspondant à des blocs. Le principe de base de cette méthode est d'effectuer des permutations des lignes et des colonnes afin de définir la structure de bloc.

L'idée de base de l'algorithme est de n'utiliser que les variables ayant une variance supérieure au seuil dans une classe donnée pour découper cette classe. Soit $A(I, J)$ une matrice de données avec $1 \leq I \leq N, 1 \leq J \leq d$. Les classes en lignes $1, 2, \ldots, K$ sont construites de manière que la classe I est déterminée par les classes qui la divisent $Min(I)$ et $Max(I)$. Pour une classe minimale (que l'on ne peut plus diviser), ces valeurs représentent la première et la dernière ligne de la classe I. À la fin de l'algorithme, $V(I)$ est défini comme l'ensemble des variables qui ont une variance inférieure au seuil dans I, et dans aucune autre classe plus grande.

L'algorithme procède par découpages de classes successifs. À l'étape p, il existe p classes $I(1), I(2), \ldots, I(p)$ séparant les lignes, qui sont les classes minimales dans l'ensemble $1, 2, 3, \ldots, 2p-1$. $V[I(J)]$ représente l'ensemble des variables avec une variance supérieure au seuil T pour toute classe plus grandes [Jollois 2003]. Un découpage en deux est effectué sur les classes $I(J)$, en utilisant seulement les variables dans $V[I(J)]$ qui ont une variance supérieure au seuil. Les deux nouvelles classes $2p$ et $2p+1$ auront $V(2p) = V(2p+1)$ défini comme l'ensemble des variables de $I(J)$ qui ont une variance supérieure à T dans $I(J)$. Aussi, $V[I(J)]$ sera changé en l'ensemble des variables de $V[I(J)]$

qui auront donc une variance inférieure à T dans $I(J)$. Le découpage s'arrête lorsque tous les ensembles $V[I(J)]$ contiennent des variables avec une variance inférieure au seuil dans $I(J)$.

1.4.5 Méthodes hiérarchiques

On retrouve dans la littérature plusieurs approches de bi-partitionnement qui utilisent des algorithmes hiérarchiques. Nous citons les travaux de [Caldas 2011], [Mao 2005] et [Getz 2000a]. Une des approches les plus utilisées dans cette famille de modèles est CTWC (Coupled Two-Way Clustering) [Getz 2000a]. CTWC consiste à appliquer un algorithme de classification hiérarchique, le SPC "Super Paramagnetic Clustering" [Getz 2000b], sur les colonnes en utilisant toutes les lignes puis sur les lignes en utilisant toutes les colonnes. Toutes les sous-matrices (I, J), sachant que I est une classe sur les lignes et J une classe sur les colonnes sont calculées. Seules les sous-matrices qui satisfont un certain critère comme la stabilité ou une taille minimale sont retenues [Charrad 2008]. Ensuite, le processus est réitéré : des classes de lignes et de colonnes sont extraites à partir de ces sous-matrices.

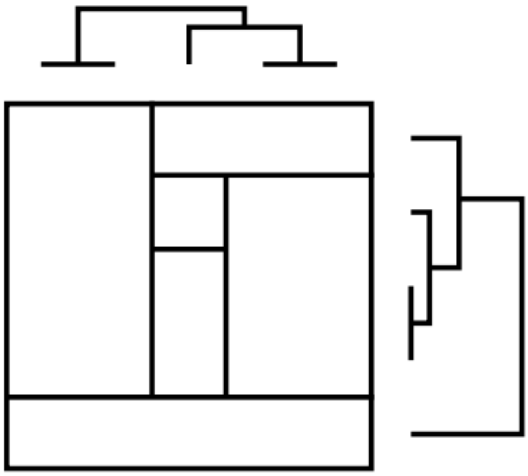


FIGURE 1.12 – Approche Two-way splitting : Recherche de blocs de données homogènes et des hiérarchies en ligne et en colonne.

CTWC opère sur l'ensemble des sous-ensembles des observations v et sur l'ensemble des sous-ensembles des variables $\{u\}$. Initialement, $\{v\} = \{V\}$ et $\{v\} = \{U\}$, l'algorithme sélectionne de manière itérative un sous-ensemble de gènes $\{V\}' \in v$, et un sous-ensemble de variables $\{U\}' \in u$; ensuite l'algo-

rithme de SPC est appliqué sur $\{V\}^{'}$ et $\{U\}^{'}$. L'algorithme correspondant est décrit comme suit :

Algorithme 7 : Algorithme CTWC

1: Entrées : $\mathcal{A}$: matrice des données.
2: Sorties : les partitions des observations v et les partitions des variables u.
3: Phase d'initialisation :
   – $v_1 = \{V\}$, $u_1 = \{U\}$, $v = \oslash$, $u = \oslash$.
   – Initialiser le tableau hiérarchique H_v pour la sauvegarde des clusters d'observations.
   – Initialiser le tableau hiérarchique H_u pour la sauvegarde des clusters de variables.
4: *Tant que* $(u_1 \neq \oslash$ *ou* $v_1 \neq \oslash)$ *faire*
5: *Pour* $(U^{'}, V^{'}) \in (u_1 \times v_1) \cup (u_1 \times v) \cup (u \times_1)$ *faire*
6: Appliquer l'algorithme SPC $(E_{U'V'})$ pour le clustering des observations $V^{'}$
   – Ajouter à l'ensemble des observations stables v_2
   – $H_V[V^{''}] = U^{'}$ pour tous les nouveaux clusters $V^{''}$
7: Appliquer l'algorithme SPC $(E_{U'V'})$ pour le clustering des observations $U^{'}$
   – Ajouter à l'ensemble des observations stables u_2
   – $H_U[V^{''}] = V^{'}$ pour tous les nouveaux clusters $U^{''}$
8: $u = u \cup u_1$, $v = v \cup v_1$
9: $u_1 = u_1$, $v_2 = v_2$
10: Retourner u, v et leur hiérarchie H_U, H_V.

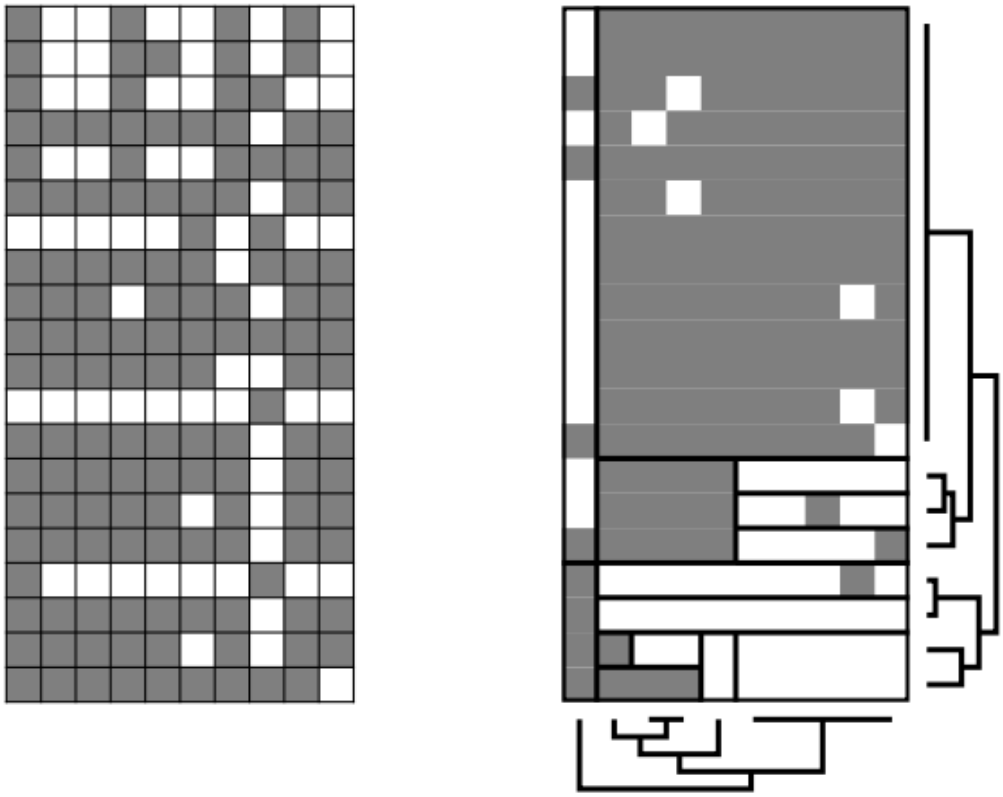


FIGURE 1.13 – Résultats d'une méthodes classification croisée hiérarchiques : CTWC

1.4.6 Méthodes constructives

Dans ce type d'approches, les blocs de données sont construits de différentes manières [Charrad 2008]. Par exemple : par ajout et suppression des lignes et des colonnes (δ-biclusters [Cheng 2000]), par permutation des lignes et des colonnes (OPSM [Ben-Dor 2002]), par estimation des paramètres des modèles (plaid models [Lazzeroni 2000]) ou à partir d'un graphe biparti (SAMBA [Tanay 2002]), etc.

L'algorithme SAMBA proposé par Tanay et al. [Tanay 2002] représente une matrice de données par un graphe G biparti pondéré où chaque noeud n_i correspond à une ligne et chaque noeud n_j correspond à une colonne. L'arête entre le noeud n_i et le noeud n_j a un poids a_i^j correspondant à l'élément de la matrice se trouvant à l'intersection de la ligne i et de la colonne j. Une biclasse correspond au sous-graphe (H, J, E) de G et représente un sous-ensemble I

d'observations dont la valeur change significativement sous un ensemble de variables J. L'objectif de l'algorithme SAMBA est de chercher dans les données des biclasses maximales. L'application de l'algorithme SAMBA est effectuée en deux étapes :

1. Les données sont normalisées et représentées par un graphe biparti,
2. L'algorithme identifie les k bi-cliques maximales.

Dans une phase ultérieure, SAMBA apporte des améliorations locales aux biclasses par ajout ou suppression des sommets, et sélectionne les biclasses similaires ayant un nombre important de sommets en commun.

L'approche δ-biclusters [Cheng 2000] est une méthode de bi-clustering constructive. Le principe de l'algorithme δ-biclusters consiste à supprimer itérativement des lignes et des colonnes à partir de la matrice initiale jusqu'à ce que la mesure de distance soit inférieure à un certain seuil, puis ajouter des lignes et des colonnes itérativement sans entraîner une augmentation de cette mesure de distance [Charrad 2008]. À chaque itération, une biclasse est générée puis remplacée dans la matrice initiale par des valeurs aléatoires. Une limite de cette approche est que le nombre de biclasses à rechercher doit être fixé par l'utilisateur tout comme le seuil δ utilisé pour la mesure de la qualité. En plus, la qualité des biclasses diminue à chaque itération à cause des valeurs aléatoires ajoutées à chaque itération.

Les auteurs de [Ben-Dor 2002] ont défini un bloc comme une sous-matrice préservant l'ordre des données. Contrairement aux méthodes d'estimation des paramètres [Lazzeroni 2000] où l'uniformité des données dans la matrice de données est considérée, ils se concentrent plutôt sur l'ordre relatif des colonnes dans les blocs. L'objectif d'OPSM est l'identification des grands blocs. Une sous-matrice est préservatrice de l'ordre s'il existe une permutation des colonnes permettant d'avoir des valeurs strictement croissantes sur chaque ligne.

1.4.7 Décomposition matricielle pour le bi-clustering

Récemment, de nouvelles approches de bi-partitionnement basées sur la factorisation matricielle sont proposées ([Long 2005], [Yoo 2010], [Labiod 2011], [Shang 2012]). Dans ce type d'approche, le problème de bi-partitionnement peut être vu comme un problème d'approximation matricielle où l'objectif est de minimiser l'erreur d'approximation entre les données de la matrice d'origine $\mathcal{A}$ et la matrice reconstruite sur la base des structures de classes. Étant donné une matrice non négative $\mathcal{A}$, la stratégie générale d'une approche de bi-partitionnement dans ce contexte consiste à rechercher une décomposition de $\mathcal{A}$ sous la forme de trois matrices $\mathbf{ZGW}^T$. La matrice

$\mathbf{Z}$ représente le partitionnement des lignes de $\mathcal{A}$, la matrice $\mathbf{W}$ représente le partitionnement des colonnes de $\mathcal{A}$ et la matrice $\mathbf{G}$ est une matrice intermédiaire. La plupart des algorithmes proposés dans ce sens sont itératifs. Seules les règles de mise à jour des trois matrices (méthode d'optimisation choisie ou contraintes imposées sur les trois matrices) peuvent être différentes.

Algorithme CUNMTF

L'approche CUNMTF (Co-clustering Under Nonnegative Matrix Tri-Factorization) de Labiod et Nadif [Labiod 2011] propose une nouvelle formulation du modèle NMF[3] [Lee 1999] adaptée au bi-partitionnement. Les auteurs proposent deux approches qui optimisent une formulation relaxée du critère des double K-means dans un style NMF. La première appelée DNMF et la second ODNMF lorsque les contraintes d'orthogonalité sur $\mathbf{Z}$ et $\mathbf{Z}$ sont considérées. L'idée principale de cette approche est que la structure du bloc latent dans une matrice de données rectangulaire non négative est factorisée en deux facteurs plutôt que trois : la matrice des coefficients des lignes $\mathbf{R}$ et la matrice des coefficients des colonnes $\mathbf{C}$, qui indiquent respectivement le degré d'appartenance d'une ligne et d'une colonne à un cluster. Les auteurs proposent d'abord une formulation du modèle double K-means, qui est appelé DNMF (Double Nonnegative Matrix Factorization).

Étant donnée une matrice $\mathcal{A} = (x_i^j) \in \mathcal{R}^{N\times d}$, le but du double K-means est de trouver simultanément une partition en K classes $P = \{P_1, \ldots, P_K\}$ de l'ensemble des lignes $I = \{1, \ldots, N\}$ et une partition $Q = \{Q_1, \ldots, Q_L\}$ en L classes de l'ensemble des colonnes $J = \{1, \ldots, d\}$. Les deux partitions P et Q induisent naturellement et respectivement des matrices de classification $Z = (z_{ik}) \in \{0,1\}^{N\times K}$ et $W = (w_{jl}) \in \{0,1\}^{d\times L}$; $z_{ik} = 1$ (resp. $w_{jl} = 1$), si la ligne $\mathbf{x}_i \in P_k$ (resp. la colonne $\mathbf{x}^j \in Q_l$), et 0 sinon.

La réorganisation des lignes et des colonnes suivant P et Q révèle une structure de blocs homogènes. Chaque bloc $\mathcal{A}_k^l$ est donc défini par $\{(x_i^j)|z_{ik}w_{jl}a_i^j = 1\}$. D'autre part, $G = (g_k^l) \in \mathcal{R}^{K\times L}$ est le représentant de taille réduite de $\mathcal{A}$ (g_k^l est le barycentre de $\mathcal{A}_k^l$).

La détection des blocs homogènes en $\mathcal{A}$ peut être obtenue par la recherche des trois matrices Z, W et G en minimisant

$$\mathcal{J}(\mathcal{A}, ZGW^T) = ||\mathcal{A} - ZGW^T||^2$$

Le terme ZGW^T caractérise l'information de $\mathcal{A}$ qui peut être décrite par une structure de classes.

3. Voir section 1.3.5

Cette formulation matricielle peut prendre la forme : suivante :

$$\mathcal{J}(\mathcal{A}, ZGW^T) = \sum_{i,j,kl} z_{ik} w_{jl} (x_i^j - g_k^l)^2$$

Avec P_k , Q_l fixées, le terme général de G optimale est obtenu par :

$$g_k^l = \frac{\sum_{i,j,k,l} z_{ik} w_{jl} x_i^j}{z_k w_l}$$

Où $z_k = |P_k|$; $w_l = |Q_l|$.

Dans le cadre du double K-means, la fonction objectif à minimiser est la distance au carrée entre chaque ligne (chaque colonne) du centre. Soit $D_z^{-1} \in \mathcal{R}^{K \times K}$ et $D_w^{-1} \in \mathcal{R}^{L \times L}$ deux matrices diagonales définies par $D_z^{-1} = Diag(z_1^{-1}, \ldots, z_K^{-1})$ et $D_w^{-1} = Diag(w_1^{-1}, \ldots, w_L^{-1})$. En utilisant les matrices D_z, D_w, $\mathcal{A}$, Z et W, la matrice de représentation G s'écrit : $G = D_z^{-1} Z^T \mathcal{A} W D_w^{-1}$. Si G est intégré dans la fonction objectif $\mathcal{J}(\mathcal{A}, ZGW^T)$, alors l'expression à optimiser devient $||\mathcal{A} - \mathbf{Z}\mathbf{Z}^T \mathcal{A} \mathbf{W}\mathbf{W}^T||^2$, où $\mathbf{Z} = ZD_z^{-0.5}$ et $\mathbf{W} = WD_w^{-0.5}$. Les auteurs affirment que cette formulation est valable même si $\mathcal{A}$ n'est pas non négative, et l'approximation $\mathbf{Z}\mathbf{Z}^T \mathcal{A} \mathbf{W}\mathbf{W}^T$ de $\mathcal{A}$ est formée par la même valeur dans chaque bloc $\mathcal{A}_k^l$. Plus précisément, la matrice $\mathbf{Z}^T \mathcal{A} \mathbf{W}$ joue le rôle d'un résumé de $\mathcal{A}$, et absorbe les différences d'échelle de $\mathcal{A}$, $\mathbf{Z}$ et $\mathbf{W}$. Les matrices $\mathbf{Z}\mathbf{Z}^T \mathcal{A}, \mathcal{A}\mathbf{W}\mathbf{W}^T$ donnent respectivement les vecteurs des moyennes des classes en ligne et en colonne.

Ensuite, les auteurs définissent le modèle CUNMTF en introduisant la fonction objectif :

$$arg \min_{\mathbf{Z},\mathbf{W} \geq 0} ||\mathcal{A} - \mathbf{Z}\mathbf{Z}^T \mathcal{A} \mathbf{W}\mathbf{W}^T||^2$$

et en prenant en compte la contrainte de non négativité. Afin d'optimiser cette fonction objectif, les auteurs utilisent les conditions de Karush-Kuhn-Tucker [Boyd 2004] en introduisant la fonction de Lagrange :

$$\mathcal{L} = ||\mathcal{A} - \mathbf{Z}\mathbf{Z}^T \mathcal{A} \mathbf{W}\mathbf{W}^T||^2 - Trace(\Lambda \mathbf{Z}^\mathbf{T}) - Trace(\Gamma \mathbf{W}^\mathbf{T})$$

où les matrices Λ et Γ sont les multiplicateurs de Lagrange introduits pour imposer la contrainte de non négativité respectivement sur $\mathbf{Z}$ et $\mathbf{W}$. Soit, $X_W = \mathcal{A}\mathbf{W}\mathbf{W}^T$ et $X_Z = \mathbf{Z}\mathbf{Z}^T \mathcal{A}$. Cela conduit aux règles de mise à jour suivantes :

$$\mathbf{Z} \leftarrow \mathbf{Z} \odot \frac{2\mathcal{A} X_\mathbf{W}^T \mathbf{Z}}{\mathbf{Z}\mathbf{Z}^T X_\mathbf{W} X_\mathbf{W}^T \mathbf{Z} + X_\mathbf{W} X_\mathbf{W}^T \mathbf{Z}\mathbf{Z}^\mathbf{T}\mathbf{Z}} \tag{1.37}$$

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{2\mathcal{A} X_\mathbf{Z}^T \mathcal{A} \mathbf{W}}{\mathbf{W}\mathbf{W}^T X_\mathbf{Z} X_\mathbf{Z}^T \mathbf{W} + X_\mathbf{Z} X_\mathbf{Z}^T \mathbf{W}\mathbf{W}^\mathbf{T}\mathbf{W}} \tag{1.38}$$

Les auteurs proposent ensuite, un algorithme pour calculer la relaxation non négative. L'algorithme contient les étapes classiques de l'approche NMF. L'estimation obtenue par cet algorithme est améliorée itérativement en mettant à jour les facteurs avec les règles 1.37 et 1.38. Pour dériver les règles multiplicatives de mise à jour sous les contraintes d'orthogonalité sur $\mathbf{Z}$ et $\mathbf{W}$, les auteurs calculent le "gradient naturel" sur les variétés de Stiefel [Freitas 1985]. Les règles de mise à jour, sont donc :

$$\mathbf{Z} \leftarrow \mathbf{Z} \odot \frac{\mathcal{A}\mathbf{W}\mathbf{W}^T\mathcal{A}^T\mathbf{Z}}{\mathbf{Z}\mathbf{Z}^T\mathcal{A}\mathbf{W}\mathbf{W}^T\mathcal{A}^T\mathbf{Z}}$$

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{\mathcal{A}\mathbf{Z}\mathbf{Z}^T\mathcal{A}\mathbf{W}}{\mathbf{W}\mathbf{W}^T\mathcal{A}^T\mathbf{Z}\mathbf{Z}^T\mathcal{A}\mathbf{W}}$$

Long et al. [Long 2005] ont proposé l'approche NBVD (Non-negative Block Value Decomposition) d'approximation matricielle de $\mathcal{A}$ basée sur une procédure itérative d'optimisation des moindres carrés alternée. A la convergence, $\mathbf{Z}\mathcal{A}$ est normalisée à $\mathbf{Z}\mathcal{A}\mathbf{X}$ ($\mathbf{X}$ est une matrice diagonale), les étiquettes des classes des colonnes, sont déduites à partir de $\mathbf{X}\mathbf{X}^{-1}\mathbf{W}^T$. Les étiquettes des classes des lignes sont déduites en travaillant sur $\mathcal{A}^T$.

1.5 Conclusion

Nous avons présenté dans ce chapitre la fouille de données, qui constitue le coeur du processus d'extraction de connaissances à partir des données. Ensuite, nous avons détaillé les techniques de classification et en particulier les techniques de classification non supervisées qui font l'objet de nos travaux. Nous avons aussi présenté quelques approches de bi-partitionnement.

La détection d'outliers est une étape incontournable de la tâche de prétraitement des données qui a un impact conséquent sur le processus d'extraction de connaissances. Plusieurs approches ont été développées dans la littérature scientifique. Cependant, rares sont les méthodes qui offrent des solutions pertinentes au sujet de la détection de "groupes-outiers". C'est-à-dire, un groupe de données qui se comporte différemment de la tendance générale d'une base de données. C'est ainsi que nous proposons une étude bibliographique sur le sujet dans le chapitre 2, et une contribution sur la détection de groupes-outliers et des nouveautés dans le chapitre 3.

Chapitre 2

État de l'art sur la détection d'outliers, de groupes-outliers et des nouveautés

Sommaire

2.1 Motivation et challenge

Définition 1 : *Un **"outlier"** est une observation ou un motif d'observations qui n'est pas conforme ou "normal"*[1] *par rapport au comportement global de l'ensemble des données.*

Définition 2 : *Un **"groupe-outlier"** est un ensemble de données formant un groupe dense et significativement isolé.*

1. Nous utilisons le mot "normal" ici en terme simple, et non comme une référence à la distribution normale dans les statistiques

Les données particulières qui ne sont pas conformes au comportement attendu sont souvent appelées des outliers, des anomalies, des événements, des données aberrantes, des données d'intérêt, des nouveautés, des exceptions, des données discordantes, etc. Cela dépend de la perception que l'on donne à ces données particulières, aussi et dans la plupart des cas, du contexte visé. Nous allons dans toute la suite de cette thèse utiliser le terme "outlier".

Étant donné la quantité croissante des données recueillies universellement, il devient à la fois plus important et difficile de repérer les observations inhabituelles (outliers) ou inattendues (nouveautés). Un tel comportement inattendu peut être soit non désiré (par exemple, la détection d'intrusion réseau, la surveillance des maladies), nécessitant une intervention de l'utilisateur, ou il peut être intéressant (en astronomie notamment), ce qui conduit à une meilleure compréhension du système. La tâche de détection d'outliers joue un rôle important, puisque, dans la plupart des cas, la détection d'outliers permet de prévenir ou d'atténuer les effets d'une situation indésirable. Par exemple, dans la bio-surveillance, il est indisponsable de détecter les épidémies qui donnent lieu à des schémas inhabituels dans les dossiers des services d'urgence. La détection précoce de ces phénomènes conduisant à des mesures appropriées peuvent sauver de nombreuses vies. Par conséquent, les systèmes de surveillance automatiques sont de plus en plus populaires et utilisent des méthodes de fouille de données pour effectuer une détection. L'observation des procédés de fabrication industrielle est une application traditionnelle de ces systèmes. Les données chronologiques provenant de différents capteurs sont surveillées afin de détecter les outliers dans les processus de contrôle. Une autre application courante est la surveillance de la santé publique, où les données des patients dans les hôpitaux et les ventes de médicaments en pharmacies sont surveillées dans le but de détecter les éclosions de maladies nouvelles le plus tôt possible [Andrew Moore 2003]. D'autres applications de détection d'outliers incluent la détection des fraude des cartes de crédit [Bhattacharyya 2011], de traitement d'image [Chen 2005, Bishop 1994] et la surveillance du trafic [Shekhar 2001].

Un défi important dans la détection d'outliers est la difficulté d'obtenir des données suffisamment marquées pour caractériser les outliers. Par conséquent, dans la plupart des cas, nous devons opérer dans un environnement non supervisé, où seul le comportement normal est caractérisé, et est utilisé pour détecter les écarts par rapport à celui-ci. Dans le cadre d'exploration des données, il est généralement admis que nous avons un ensemble de données d'apprentissage suffisamment grand qui ne contient pas ou très peu de cas anormaux. Cet ensemble de données est supposé définir le comportement normal du système. Parallèlement à cela, nous avons également besoin d'une mesure (score) "d'outlier-ness" (aussi appelée "d'anomalous-ness" ou "d'aberrance"), qui permet de comparer les nouvelles observations aux données initiales. Compte

tenu de cette méthode de notation, toute observation qui s'écarte de manière significative de l'habituel est signalée comme une donnée outlier ou nouvelle. Cela a conduit à la spécification du type d'outliers, la nature des données, la disponibilité des étiquettes des données et d'autres contraintes. À la lumière de ces facteurs, nous présentons la relation entre les différentes techniques de détection d'outliers dans la suite de ce chapitre.

2.2 Détection d'outliers et de groupes-outliers

Les données sont généralement un ensemble d'observations, chacune d'elles étant décrite par un ensemble d'attributs (variables ou caractéristiques). D'une manière générale, les outliers peuvent être soit des outliers individuels (correspondant à une seule observation) ou des groupes-outliers, aussi appelés "outliers collectifs" (correspondant à des groupes d'observation). Dans le cas de la détection d'outliers individuels, une approche standard consiste à créer un modèle de données normales, et de comparer les observations de la base de test. Une approche probabiliste construit un modèle de vraisemblance à partir des données d'apprentissage. Dans le cas des groupes-outliers, plutôt que de trouver des comportements individuels anormaux (ce qui peut être dû à un bruit ou à des erreurs dans les données), nous sommes plus intéressé par la détection de l'émergence de nouveaux phénomènes résultant des modes d'observation anormaux qui ne peuvent être expliqués par un précédent modèle. Notre objectif ici est d'utiliser la présence de ces multiples cas afin de mieux détecter les groupes de données anormaux que nous appelons groupes-outliers. Notre contribution dans le chapitre suivant est dédiée plutôt à la détection des "groupes-outliers".

Dans certains cas, les attributs peuvent être divisés en deux ensembles distincts, les attributs contextuels et les attributs comportementaux [Song 2007]. Les attributs contextuels précisent le contexte et les attributs comportementaux déterminent si oui ou non les observations sont anormales dans un contexte donné. La plupart des travaux antérieurs qui visent à détecter les groupes-outliers supposent une certaine forme d'information contextuelle. Dans ce cas, la définition d'un groupe repose sur la similarité entre les observations à l'égard de ces attributs contextuels. Par exemple, en mode de balayage spatial, un groupe est défini comme un ensemble d'emplacements géographiquement adjacents. S'il l'on traite une région géographique spécifique, le nombre de patients atteints d'un symptôme particulier (attribut comportemental) détermine si oui ou non une épidémie est survenue dans la région [Kaustav Das 2008, Agarwal 2006].

Une caractéristique importante d'une méthode de détection d'outliers est

la disponibilité (ou non disponibilité) des étiquettes de données. Dans la pratique, l'obtention des étiquettes appropriées (chaque observation de données est normale ou anormale) est une tâche difficile et fastidieuse. Souvent, la seule source fiable est une expertise humaine pour étiqueter les données. Il existe trois catégories de méthodes : supervisées, semi supervisées et non supervisées.

Les méthodes supervisées supposent que nous disposons d'un ensemble de données d'apprentissage entièrement étiquetées avec des étiquettes de classe à la fois normales et anormales. Habituellement, ceci est utilisé pour former une méthode de classification appropriée, qui est ensuite utilisée pour classer les données de la base de test. Dans la famille des approches semi supervisées, nous disposons de quelques étiquettes, qui proviennent soit des classes normales soit des classes anormales (ou dans certains cas des deux classes). Les données d'apprentissage se composent généralement de ces étiquettes avec un grand nombre de cas non étiquetés. La famille des approches non supervisées suppose l'absence de toutes les étiquettes dans les données d'apprentissage. Cependant, dans la plupart des cas, les bases d'apprentissage contiennent beaucoup plus d'étiquettes normales que d'étiquettes anormales. Ainsi, un modèle de comportement normal peut être appris à partir des données d'apprentissage.

Dans de nombreux cas, il est important de détecter un comportement inattendu ou inexpliqué qui ne peut pas être pré-spécifié ou prédit. D'où la nécessité des approches non supervisées qui reposent sur la détection de toutes les observations qui s'écartent significativement des données normales, elles ne sont pas limitées à un type d'outlier.

Dans cette thèse, une des principales contributions est celle de la détection des "groupes-outliers" (aussi appelés "anomalies collectives") dans un cadre non supervisé en utilisant la topologie entre les données.

2.3 Méthodes de détection d'outliers

Comme l'indique Viviane Planchon dans sa thèse [Planchon 2007], l'évolution dans la manière d'appréhender le problème du traitement des outliers est très nette. En 1852, Peirce, le premier auteur à s'intéresser au problème des valeurs anormales, disait, de manière très naïve et restrictive : *"Dans presque toutes les séries de données, il y a des observations qui diffèrent tellement des autres, qu'elles servent uniquement à rendre l'expérimentateur perplexe et à l'induire en erreur."* Les outliers n'induisent pas forcément en erreur, elles ne sont pas forcément mauvaises ou erronées. Dans certains cas, l'expérimentateur peut même être tenté de ne pas rejeter l'outlier mais de l'accepter comme

une indication intéressante. Il n'est pas approprié d'adopter une attitude radicale, soit de rejet, soit d'inclusion systématique des outliers. La première attitude peut entraîner la perte d'informations réelles, tandis que, dans le second cas de l'acceptation des outliers, il y a un risque de contamination. En fonction des circonstances, une variété de méthodes dans la littérature est proposée afin de tenir compte de toutes les données mais aussi de minimiser l'influence des outliers.

2.3.1 Méthodes basées sur la distance et la densité des données

De nombreux travaux proposent différentes méthodes pour détecter des outliers dans des données multivariées sans connaissance a priori de la distribution. Knorr et Raymond [Knorr 1997] donnent leur propre définition d'un outlier basée sur la distance. Un point est appelé un $DB(p, D)$ outlier si au moins une fraction p des points de l'ensemble de données sont à une distance supérieure à D. Ils prouvent aussi que leur définition est compatible avec les définitions d'outlier basées sur des distributions connues a priori [Plantevit 2007]. Ils définissent plusieurs algorithmes pour extraire des outliers basés sur la distance. Ramaswamy et al. [Ramaswamy 2000] montrent que les DB outliers sont trop sensibles aux paramètres p et D. Ils définissent les outliers basés sur les k plus proches voisins. Ils calculent, pour chaque donnée, les k plus proches voisins. Ils ordonnent les données par rapport à cette distance et extraient les n données les plus déviantes.

Dans les travaux de [Gao 2010], une approche appelée "noyau local multi-échelle de régression" est introduite, cette méthode permet de transformer le problème classique de détection d'outliers en un problème d'apprentissage de régression non paramétrique. Pour mettre en place cette méthode, les auteurs introduisent la notion des k-ppv. Cette méthode permet de classer les observations non étiquetées sur la base de leur similarité avec les observations de la base d'apprentissage. L'algorithme k-ppv nécessite seulement : un entier k, une base d'apprentissage et une métrique pour la proximité. En se basant sur cette logique, les auteurs de [Fabrizio 2002] ont proposé une méthode qui a comme principe d'attribuer à chaque point un poids. Ce poids représente la somme des distances de ses k-ppv. Les outliers sont les points ayant les plus grandes valeurs de poids. Pour calculer ces poids, ils cherchent les k-ppv de chaque point d'une manière rapide et efficace en linéarisant l'espace de recherche à travers la courbe de remplissage de l'espace de Hilbert. L'algorithme se compose de deux phases, la première fournit une solution approchée et la deuxième retourne la solution exacte.

Concernant les approches basées sur la densité des données, ce sont des

méthodes qui utilisent la notion de densité des observations pour la détection d'outliers. L'approche Local Outlier Factor (LOF) [Breunig 2000] apparue dans les années 2000 reste la plus utilisée dans ce type de modèles. L'avantage de cette méthode est qu'elle ne fait aucune hypothèse sur la distribution des données. Les auteurs de [Hasan 2009] ont donné une définition simplifiée de l'approche LOF. En effet, cette méthode consiste à comparer la densité locale d'une observation avec la densité moyenne de ses k-plus proches voisins locaux. La valeur du LOF est calculée après avoir défini les k-plus proches voisins locaux, la densité locale et la densité relative de chaque donnée. Soit $\mathcal{A}$ l'ensemble des données $\mathbf{x}_i$ d'apprentissage, de taille N, où chaque observation $\mathbf{x}_i = (x_i^1, x_i^2, ..., x_i^j, ..., x_i^d) \in \Re^d$. L'approche LOF associe à chaque observation $\mathbf{x}_i$ un score. Elle est constituée de 4 phases définies dans l'algorithme 8 :

Algorithme 8 : Algorithme LOF

1: **Entrée :** l'ensemble des données $\mathcal{A}$.

2: **Sortie :** score LOF associé à chaque observation $\mathbf{x}_i$.

3: Pour chaque observation $\mathbf{x} \in \mathcal{A}$, définir son voisinage local avec un minimum de points k :

$$Ne(\mathbf{x}, k) = \{\mathbf{y} \in \mathcal{A}, distance(\mathbf{x}, \mathbf{y}) \leq distance(\mathbf{x}, \mathbf{x}_k)\}$$

Où chaque $\mathbf{x}_k$ est le k^e plus proche voisin de $\mathbf{x}$. Donc $Ne(\mathbf{x}, k)$ contient au moins k points.

4: Calculer la densité locale de chaque observation :

$$density(\mathbf{x}, k) = \frac{|Ne(\mathbf{x}, k)|}{\sum\limits_{\mathbf{y} \in Ne(\mathbf{x},k)} distance(\mathbf{x}, \mathbf{y})}$$

5: Calculer la densité relative de chaque observation :

$$ard(\mathbf{x}, k) = \frac{density(\mathbf{x}, k)}{\frac{\sum\limits_{\mathbf{y} \in Ne(\mathbf{x},k)} density(\mathbf{y},k)}{|Ne(\mathbf{x},k)|}}$$

6: Déduire la valeur du LOF :

$$LOF(\mathbf{x}, k) = ard(\mathbf{x}, k)^{-1} = \frac{\frac{\sum\limits_{\mathbf{y} \in Ne(\mathbf{x},k)} density(\mathbf{y},k)}{|Ne(\mathbf{x},k)|}}{density(\mathbf{x}, k)}$$

Plus la valeur de LOF est significativement plus grande que 1, plus l'obser-

vation **x** est potentiellement outlier. À titre d'exemple, on a appliqué l'algorithme 8 sur la base Iris. La figure 2.5 montre une projection de la base Iris sur un espace 2D à l'aide de l'ACP en indiquant par une échelle de couleur les valeurs de LOF estimées pour chaque point (bleu : très faible, rouge : très forte). On remarque que plus le point est isolé, plus il est considéré comme outlier (la

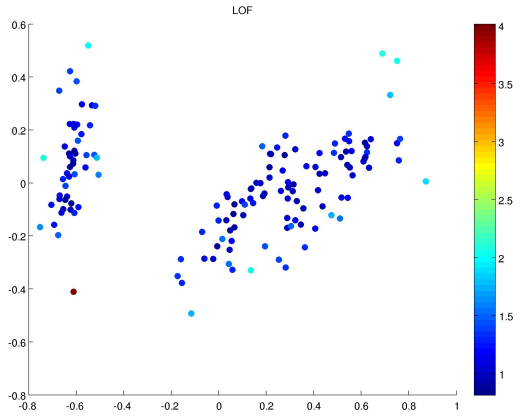


FIGURE 2.1 – Calcul des valeurs LOF avec la base des Iris.

couleur indiquée en rouge). LOF permet de mesurer "l'outlier-ness" de chaque donnée "sans apprentissage". Cette méthode est un moyen efficace pour l'identification des outliers, dans le cas où l'on se base sur la densité des données. Les auteurs de [Zengyou 2003] ont utilisé LOF au niveau des clusters pour donner de l'importance aux données au niveau local. Le modèle utilisé permet d'affecter une mesure pour chaque cluster afin d'identifier les outliers. D'autres variantes plus récentes de l'algorithme LOF continuent toujours d'apparaître dans la littérature scientifique [Mennatallah Amer 2012, E. Schubert 2012].

Aggarwal et Yu [Aggarwal 2001] assurent que les approches basées sur la distance et la densité locale ne fonctionnent pas bien dans des ensembles contenant de nombreuses dimensions, puisque les données sont "creuses" et les outliers doivent être définis dans une projection dans un sous-espace (sub space projection). Ils proposent un algorithme évolutif pour détecter les outliers.

2.3.2 Méthodes basées sur les séries chronologiques

La détection d'outliers dans les séries chronologiques représente un axe important pour les modèles statistiques appropriés, c'est ainsi que les auteurs de [Box 1965] ont étudié le changement de niveau dans les séries chronologiques. Le changement de la variance du premier ordre des modèles auto-régressifs

a été examiné par [Wichern 1976]. Ces auteurs ont étudié l'effet du changement de variance sur l'estimation des paramètres et ont suggéré la vérification d'un changement de variance en examinant les écarts résiduels. L'auteur de [Tsay 1988] a proposé une approche basée sur la technique des moindres carrés et les ratios de la variance résiduelle.

Une méthode simple de surveillance des séries temporelles consiste à placer une restriction sur les valeurs minimales et maximales (intervalle de confiance à trois écarts types par exemple) et de déclencher une alarme si le signal se situe en dehors de cet intervalle. Pour sa simplicité et son efficacité, une des méthodes les plus utilisées dans ce contexte est appelée Somme Cumulée (CUSUM) [Montgomery 2007]. Comme son nom l'indique, CUSUM maintient une somme cumulée des écarts par rapport à une valeur de référence. Considérons une série chronologique X_i, $i = 1...n$. Le calcul des sommes cumulées se fait comme suit :

$$\begin{cases} C_0 = 0 \\ C_m = max(0, X_m - (\mu_0 + K) + C_{m-1}) \end{cases} \tag{2.1}$$

μ_0 est la valeur attendue.

À partir des équations ci-dessus, si les valeurs de X_m sont proches de la moyenne, alors la valeur de C_m sera une valeur faible. Cependant, lorsqu'un changement positif, par rapport à la moyenne, se produit, la valeur de C_m augmente rapidement. K étant la valeur de la marge. Dans l'équation 2.1, les valeurs des K unités de μ_0 sont ignorées et provoquent aussi des C_m à dériver vers zéro durant le déroulement normal du système. Les alertes sont déclenchées à chaque fois que C_m dépasse un seuil de décision d'intervalle H, et C_m est remis à zéro.

Afin de procéder à une meilleure modélisation des outliers dans les séries chronologiques, les données observées peuvent être utilisées pour prédire les valeurs futures. S'il existe un écart important entre les valeurs prédites et les valeurs observées, les données sont notées comme outliers. La technique la plus courante consiste à modéliser les séries chronologiques comme une moyenne mobile autorégressive (ARMA, ARIMA) ou saisonnière (SARIMA) [Box 1990]. Un résumé de ce type de techniques peut être observé dans [Weng-Keen Wong 2002]. Les auteurs de l'approche AWSMM (Arbitrary-Window Stream Modeling Method) [Papadimitriou 2003] proposent un modèle non supervisé utilisant des coefficients d'ondelettes pour la modélisation des séries chronologiques avec des structures périodiques.

Il existe dans la littérature scientifique une classe de méthodes qui modélisent les données d'une série temporelle d'un système dynamique en utilisant l'espace continu évalué sur des états cachés. Le filtres de Kalman [Kalman 1960] supposent un système dynamique linéaire avec un bruit gaus-

sien. Une autre classe de méthodes connues sous le nom "d'algorithmes d'identification des sous-espaces" [Favoreel 1998] a pour but de déterminer directement la séquence d'états cachés sans connaître le modèle, en utilisant des outils d'algèbre linéaire comme la décomposition en valeurs singulières. Dans le cas des systèmes non linéaires avec bruit non gaussien, les filtres à particules [Gordon 2012] sont employés en utilisant des méthodes de Monte-Carlo séquentielles pour estimer le modèle. Sampling Importance Resampling (SIR) et Sampling Importance Sampling (SIS) sont d'autres techniques courantes utilisées dans cette optique [Liu 1998]. D'autres approches consistent à utiliser des arbres de décision afin de calculer la fréquence d'un motif. Cela permet de soustraire des sous-séquences dans le but de détecter les tendances anormales de la base de données [Patel 2011]. Les réseaux de neurones aussi sont utilisés pour la détection d'outliers dans les séries chronologiques [Borisyuk 2004a].

De même que l'analyse temporelle, l'analyse spatiale des données est un domaine important dans la détection des outliers, en particulier la détection de clusters spatiaux aussi appelés "bosses". Parmi le large éventail des méthodes proposées pour tester les regroupements spatiaux, la méthode de balayage spatial est une approche commune [Achtert 2011].

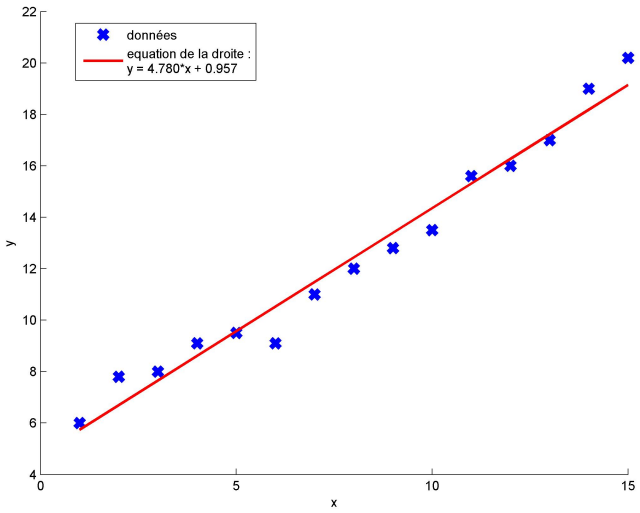


FIGURE 2.2 – Régression linéaire sans outliers sur une série chronologique.

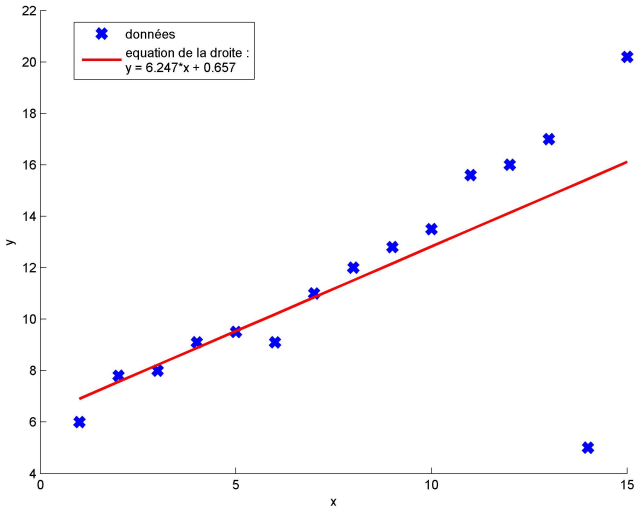


FIGURE 2.3 – Régression linéaire avec un outlier sur une série chronologique.

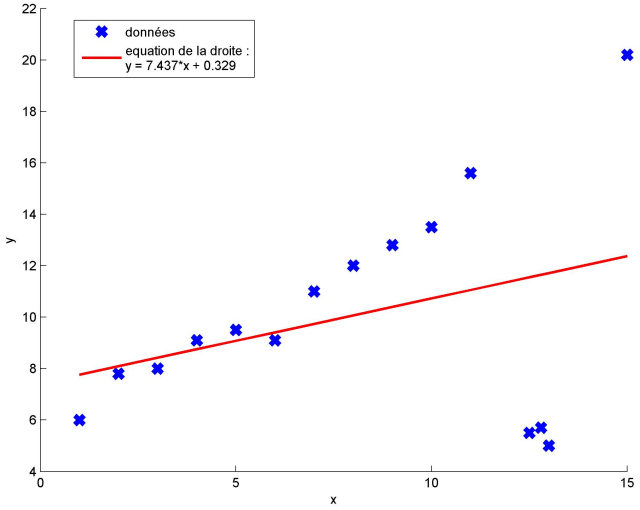


FIGURE 2.4 – Régression linéaire avec un groupe-outlier (groupe de 3 outliers) sur une série chronologique.

Nous remarquons à travers la figure 2.2 que lorsque aucun outlier n'est présent dans une série chronologique, alors la méthode de régression linéaire donne une droite représentative de l'ensemble des données. Cependant, si un outliers est présent dans la base de données, tel que dans la figure 2.3, la

droite de régression change de pente et les résultats commencent à être non représentatif de l'ensemble des données. Enfin, si un groupe-outlier est présent dans la série chronologique, comme dans l'exemple de la figure 2.4, alors les résultats de la droite de régression sont totalement biaisés. Finalement, la présence des outliers dans les bases de données de type séries chronologiques peut biaiser significativement les résultats d'une approche.

2.3.3 Méthodes basées sur les statistiques

Les statisticiens se sont intéressés à ce genre de problématiques afin de rendre les modèles mieux adaptés à leurs besoins. Les approches basées sur la distribution des données sont considérées comme les plus anciennes méthodes dans ce domaine. Elles se basent essentiellement sur les modèles statistiques (boîte à moustache, loi normale, etc.). Les premiers travaux sur la détection d'outliers proviennent du monde des statistiques où de nombreuses approches ont été développées comme les tests de discordance [Hawkins 1980, Barnett 1978]. En pratique, une règle 3σ est généralement adoptée. La règle 3σ est la suivante : soient μ la moyenne et σ l'écart type, si une observation ne se situe pas dans l'intervalle $[\mu - 3\sigma, \mu + 3\sigma]$, alors on dit que cette observation est un outlier. Certains suggèrent d'utiliser la médiane au lieu de la moyenne et de l'écart type afin de détecter des outliers multiples. Cependant, ces approches ont été développées pour extraire des outliers dans un ensemble univarié où les éléments sont supposés suivre une distribution standard (normale, poisson) alors que l'essentiel des données issues du monde réel sont multivariées et qu'il est difficile de définir la distribution qui les régit.

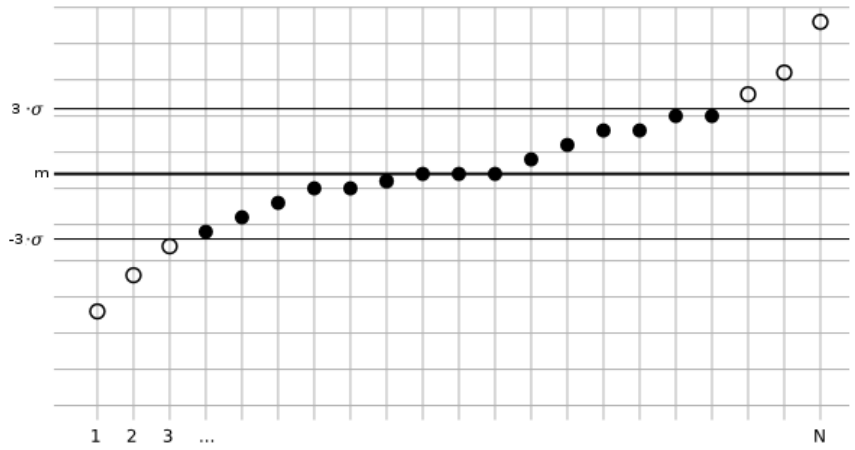


FIGURE 2.5 – Règle 3σ pour la détection d'outliers : 6 outliers détectés.

2.3.4 Autres méthodes de détection d'outliers

La caractérisation des outliers dans les séquences d'observation a été abordée par plusieurs auteurs. En effet, Sun et al. [Sun 2006] proposent d'extraire des outliers dans des bases de données séquentielles. Pour approximer les mesures de distance, ils s'appuient sur des arbres probabilistes post-fixés. Knorr et Raymond [Knorr 1998] ont proposé une version OLAP qui permet d'extraire des cellules outliers. Sarawagi et al. [Sarawagi 1998] ont proposé une exploration guidée par la découverte. Leur but est de découvrir des outliers dans les cellules du cube. Ils définissent une cellule comme outlier si la mesure (agrégat) de la cellule diffère significativement de la valeur attendue. L'écart type peut être également estimé grâce à leur proposition. Quand la différence entre la cellule et la valeur attendue est supérieure à 2,5 fois l'écart type, la cellule est considérée outlier. Leur méthode peut donc être vue comme une version OLAP de la règle 3σ.

Les auteurs de [Marascu 2009] proposent une approche Détection d'Outliers par les Ondelettes (DOO). Cette méthode n'utilise aucun paramètre et permet l'extraction automatique d'outliers dans les résultats d'un algorithme de clustering. Dans un flot de données, les données sont générées à une vitesse et dans des quantités qui interdisent toute opération bloquante. Dans ce contexte, demander un paramètre tel que k, pour les $top-k$ outliers, ou x, un pourcentage de clusters en queue de distribution, doit être évité. Premièrement, parce que l'utilisateur n'a pas assez de temps pour tester plusieurs paramètres. Deuxièmement, parce qu'une valeur choisie à un instant t dans le flot sera probablement inadaptée au temps $t+n$. En effet, d'une fenêtre d'observation sur le flot à l'autre, les résultats de la segmentation évoluent et la distribution des clusters change, ainsi que le nombre ou le pourcentage d'outliers. Cette approche, se base sur une analyse de la distribution des clusters, après les avoir triés par taille croissante. L'idée est d'utiliser la transformée en ondelettes [Young 1993] de cette distribution pour trouver la meilleure séparation. Formellement, cette transformation est définie par :

$$f(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} \psi(x)^* \left(\frac{x-b}{a} \right) dx \tag{2.2}$$

Où $\psi^*(x)$ est l'ondelette, $a(>0)$ est le facteur de mise à l'échelle et b est le paramètre de translation. Les deux coefficients les plus significatifs sont gardés et les autres sont mis à zéro.

Les auteurs de [Boudjeloud 2005] ont utilisé les algorithmes génétiques (AG) pour la détection d'outliers. En fait, l'espace de recherche du problème est représenté par une collection d'individus. Chaque individu est représenté par un tableau de caractères, chaque case est appelée "chromosome". L'objectif

est de trouver un individu avec la meilleure identité génétique de l'espace de recherche. La qualité de chaque individu est mesurée avec une fonction objectif. Une partie de l'espace de recherche est examinée à chaque itération de l'algorithme, cette dernière est appelée "population". L'AG commence par une population initiale choisie aléatoirement, et la qualité de chaque individu est évaluée. À chaque itération, deux parents sont sélectionnés, un croisement sera opéré sur les deux individus pour créer deux enfants, l'un d'eux sera remis dans la population. L'AG s'arrête après un nombre maximum d'itérations, ou après un nombre maximum de croisements ou de mutations, sans amélioration de la solution. Les individus sont constitués à partir d'un tableau contenant toutes les dimensions (attributs) disponibles qui décrivent l'ensemble des données. La procédure de recherche des données outliers suit les deux étapes suivantes :

1. Déterminer le centre de gravité de l'ensemble des données correspondant au sous-ensemble d'attributs.
2. Calculer la distance entre chaque élément et le centre de gravité de l'ensemble des données. Cette procédure aura comme sortie l'élément le plus éloigné des autres en ne considérant que le sous-ensemble d'attributs.

2.4 Détection des nouveautés

Définition 3 : *Une **"nouveauté"** est une donnée qui n'était pas connue dans la base d'apprentissage et qui apparaît dans la base de test. Le meilleur synonyme du terme "nouveauté" dans cette thèse est "inattendu".*

Il existe une forte synergie entre la détection d'outliers et la détection des nouveautés [Markou 2003a]. Ces dernières années, la détection des nouveautés a attiré une attention particulière dans le domaine de l'exploration des données en raison des nombreuses applications. Nous citons à titre d'exemple la découverte d'activités criminelles dans le commerce électronique. Dans cette section, nous donnons un bref état de l'art lié à la détection des nouveautés. La détection des nouveautés est l'identification des données nouvelles ou inconnues. Les modèles de détection des nouveautés ne dépendent pas seulement du type de méthode utilisée, mais également des propriétés statistiques des données traitées.

Il existe plusieurs approches dans la littérature scientifique qui traitent le problème de la détection des nouveautés. Par ailleurs, la détection des nouveautés reste un problème difficile à résoudre, car souvent, les méthodes proposées sont fortement sensibles aux distibutions statistiques des données à traiter [Markou 2003a].

2.4.1 Approches statistiques

L'objectif principal des approches statistiques est la modélisation des distributions des données et l'estimation de la probabilité de ces dernières d'appartenir à une distribution. Cela induit une large sensibilité des modèles par rapport aux données [Markou 2003a]. Dans les approches statistiques pour la détection des nouveautés, on se base principalement sur la modélisation des données en fonction de leurs propriétés statistiques, et on utilise ces informations pour tester si un échantillon provient de la population ou non. Les techniques utilisées varient en fonction de leur complexité. Il existe deux méthodes pour l'estimation de la densité de probabilité, les méthodes paramétriques et les méthodes non paramétriques. Les approches paramétriques supposent que les données proviennent d'une famille de distributions connues, telles que la distribution normale. Certains paramètres sont calculés pour s'adapter à cette distribution. Cependant, dans la plupart des situations du monde réel, la distribution des données n'est pas connue. L'une des techniques intuitives les plus utilisées dans les modèles non paramétriques est l'analyse de l'histogramme. La difficulté apparaît lorsqu'il s'agit d'estimer la densité des données multidimensionnelles. Dans ce cas de figure, une façon d'estimer la fonction de densité est l'algorithme de k-plus proches voisins [Odin 2000].

Les approches le plus couramment utilisées dans les statistiques [Campbell 1980, Huber 2009, Candès 2011] pour la détection des nouveautés proposent un estimateur robuste de la matrice de covariance $\mathbf{S}^*$. La moyenne et la matrice de covariance robuste peuvent être calculées comme suit :

$$\mu = \frac{\sum_{i=1}^{n} w_1(M_i^2) y_i}{\sum_{i=1}^{n} w_1(M_i^2)} \tag{2.3}$$

$$\mathbf{S}^* = \frac{\sum_{i=1}^{n} w_2(M_i^2)(y_i - \mu)(y_i - \mu)^T}{\sum_{i=1}^{n} w_2(M_i^2) - 1} \tag{2.4}$$

Où $w_1(M_i^2)$ et $w_2(M_i^2)$ sont des poids scalaires, qui sont une fonction de la distance de Mahalanobis :

$$M_i^2 = (y_i - \mu)^T \mathbf{S}^* (y_i - \mu) \tag{2.5}$$

$\mathbf{S}^*$ est estimée itérativement. De nombreuses fonctions de poids ont été proposées. Par exemple, les coefficients de pondération de Huber [Huber 2009] où $w_2(M_i^2) = (w_1(M_i^2))^2$ [Campbell 1980].

2.4.2 Approches basées sur l'ACP

L'ACP est une technique de projection orthogonale linéaire qui projette les observations multidimensionnelles représentées dans un espace de dimension d sur un sous-espace de dimension inférieure ($l < d$) en maximisant la variance des projections [Pearson 1901]. L'estimation des paramètres du modèle ACP est effectuée par calcul des valeurs et vecteurs propres de la matrice de corrélation des données [2].

Soit un tableau de données $\mathcal{A}$ $(n \times d)$ formant un nuage de n points dans un espace à d dimensions. L'ACP consiste à projeter les points sur une droite, un plan, un sous-espace, ... à l dimensions (avec $l \leq d$) choisi de façon à optimiser un certain critère. Intuitivement, on cherchera le sous-espace donnant la meilleure visualisation possible de notre nuage de points. Un bon choix consiste à rechercher la plus grande dispersion (le plus grand étalement) possible des projections dans le sous-espace choisi. On est amené ainsi à chercher une rotation de notre système d'axes initial (les variables), permettant de mieux voir notre nuage [Boubou 2003].

L'espace orthogonal défini par l'ACP est engendré par les vecteurs propres associés aux valeurs propres λ_a de la matrice de corrélation Σ de $\mathcal{A}$. Considérons $\mathbf{x} \in \mathcal{R}^d$ un vecteur de données aléatoires constitué de d variables. Soit la matrice de données $X \in \mathcal{R}^{n \times d}$ de vecteurs lignes $\mathbf{x}_i^t$ qui rassemble les n mesures sur les d variables. L'ACP détermine une transformation optimale (vis-à-vis d'un critère de variance) et linéaire de la matrice de données X comme suit :

$$X = EP^t \tag{2.6}$$

avec $E = [e_1 e_2 \dots e_d] \in \mathcal{R}^{n \times d}$ est l'ensemble des composantes principales et $P = [p_1 p_2 \dots p_d] \in \mathcal{R}^{d \times d}$ représente la matrice des vecteurs propres associés aux valeurs propres λ_a de la matrice de corrélation Σ de X :

$$\Sigma = P\Lambda P^t \; avec \; PP^t = P^t P = I_d \tag{2.7}$$

où $\Lambda = diag(\lambda_1 \dots \lambda_d)$ est une matrice diagonale dont les éléments sont mis dans l'ordre décroissant. Puisque l'objectif de l'ACP est de réduire la dimension de l'espace, les l premières composantes principales ($l << d$) sont les plus significatives et suffisent pour expliquer la variabilité d'un processus à travers sa base de données X. Par conséquent, la partition en vecteurs propres et composantes principales donne respectivement :

$$P = [\hat{P}_l | \tilde{P}_{d-l}], \; E = \hat{E}_l | \tilde{E}_{d-l}] \tag{2.8}$$

2. On s'est largement inspiré dans cette section des travaux de [Baligh 2012]

Les l premiers vecteurs propres constituent le sous-espace de representation ou le sous-espace principal (S_p) défini par : $S_p = span\{\hat{P}_l\}$. Alors que le sous-espace résiduel (S_r), est décrit par : $S_r = span\{\tilde{P}_{d-l}\}$. Ces deux sous-espaces, S_p et S_r sont orthogonaux. Un vecteur d'observation $\mathbf{x}$ se projette sur le nouvel espace et se décompose sur les deux sous-espaces S_p et S_r respectivement comme suit :

$$\hat{\mathbf{x}} = \hat{P}_l \hat{P}_l^t \mathbf{x} = \tilde{C}_{\mathbf{x}} \in S_p \tag{2.9}$$

$$\tilde{\mathbf{x}} = \tilde{P}_{d-l} \tilde{P}_{d-l}^t \mathbf{x} = \tilde{C}_{\mathbf{x}} \in S_r \tag{2.10}$$

$$\hat{\mathbf{x}}^t \tilde{\mathbf{x}} = \tilde{\mathbf{x}}^t \hat{\mathbf{x}} = 0 \; et \; \mathbf{x} = \hat{\mathbf{x}} + \tilde{\mathbf{x}} \tag{2.11}$$

où $\hat{\mathbf{x}}$ et $\tilde{\mathbf{x}}$ sont respectivement la projection de $\mathbf{x}$ sur les deux sous-espaces S_p et S_r engendrés respectivement par les l premières composantes principales et les $(d-l)$ restantes. $\hat{C}$ et $\tilde{C} = (I - \hat{C})$ représentent les matrices de projection respectivement sur les sous-espaces S_p et S_r.[3]

L'ACP classique est fondée sur l'hypothèse que les données sont toutes normales (pas de nouveautés). Or, dans la pratique, les données réelles contiennent souvent des données nouvelles (qui n'étaient pas connues lors de la phase d'apprentissage), et habituellement, elles ne sont pas faciles à séparer de l'ensemble des données. Plusieurs approches ont été proposées afin d'apporter une modélisation de la détection des nouveautés en se basant sur l'ACP ([Jolliffe 1986, ling Shyu 2003, Hoffmann 2007, Ringberg 2007, Mahmoud 2012, Lee 2013]).

En effet, après la construction du modèle ACP et pour tester une nouvelle observation, cette dernière sera projetée sur le nouvel espace. Elle sera caractérisée par une première distance, notée "$T2$ de Hotelling", dans le sous-espace S_p et une seconde, appelée "SPE" : "*squared prediction error*", dans le sous-espace S_r. Ces deux distances sont utilisées pour la surveillance et le suivi du processus. La statistique de Hotelling mesure la variation dans le sous-espace S_p. Elle est exprimée comme suit :

$$T2 = \hat{e}^t \hat{\Lambda}^{-1} \hat{e} = \sum_{a=1}^{l} \frac{e_a^2}{\lambda_a} \tag{2.12}$$

où $\hat{e} = \hat{P}^t \mathbf{x}$.

Généralement, pour un processus sous contrôle possédant des données qui suivent une distribution multi-normale, la distance $T2$ peut être approchée par une distribution de Khi-deux. Ainsi, le système est normal si :

$$T2 \leq \chi^2_{l,1-\alpha} \tag{2.13}$$

3. Le lecteur désirant de plus amples détails sur l'ACP pourra consulter [Saporta 1990] chapitre 8 section 2.

avec l degrés de liberté et $(1-\alpha)\%$ représente le quantile de la distribution. Un changement au niveau des variables corrélées indique une situation exceptionnelle car ces variables ne conservent pas leurs relations normales. Ainsi, l'observation $\mathbf{x}$ augmente sa projection sur le S_r. En conséquence, l'amplitude de $\tilde{\mathbf{x}}$ atteint des valeurs anormales comparées à celles obtenues durant les conditions normales. Le critère SPE est l'amplitude de $\tilde{\mathbf{x}}$ ainsi son expression est donnée par :

$$SPE = \| \hat{\mathbf{x}} \|^2 = \| \tilde{C}_{\mathbf{x}} \|^2 = \hat{e}^t \tilde{e} = \sum_{a=1+1}^{d} e_a^2 \tag{2.14}$$

où $\tilde{e} = \tilde{P}^t \mathbf{x}$.

Le processus est considéré normal si la statistique SPE ne dépasse pas une limite de contrôle donnée par [ling Shyu 2003, Hubert 2003] :

$$SPE \leq \delta^2_{1-alpha} = (\hat{\mu} + \hat{\sigma} z_{1-\alpha})^3 \tag{2.15}$$

avec $z_{1-\alpha} = \Phi^{-1}(1-\alpha)$ en tant que $(1-\alpha)\%$ quantile de la distribution gaussienne Φ. $\hat{\mu}$ et $\hat{\sigma}$ sont respectivement la moyenne et l'écart-type estimés de $SPE^{\frac{2}{3}}$ calculé à partir des données utilisées pour la construction du modèle ACP.

2.4.3 Approches basées sur les SVM

Les SVM (machines à vecteur support) sont une classe de techniques d'apprentissage introduite par Vladimir Vapnik au début des années 90 [Vapnik 1995]. Elles reposent sur une théorie mathématique solide, à l'inverse des méthodes de réseaux de neurones. Les SVMs sont dans leur origine utilisées pour la classification binaire et la régression. Aujourd'hui, elles sont utilisées dans différents domaines de recherche et d'ingénierie, tels que le diagnostic médical, le marketing, la biologie, la reconnaissance de caractères manuscrits et de visages humains.

Les "One-class SVM" sont une variante des machines à vecteur support où nous disposons des observations avec des étiquettes positives et d'autres négatives. De telles informations ne sont pas souvent disponibles dans tous les cas d'application. Parfois, il est très coûteux, voire impossible, de trouver les classes négative. Plusieus approches ont été développés pour la détection des nouveautés en se basant sur les SVM [Tong 2002, Zhang 2004, Hamel 2009, Winter 2011]. Pour la classification One-class SVM, il est supposé que seules les données de la classe cible sont disponibles. L'objectif est de trouver une frontière qui sépare les observations de la classe cible du reste de l'espace,

autrement dit, une frontière autour de la classe cible qui accepte autant d'observations cibles que possible [Hamel 2009]. Cette frontière est représentée par une fonction de décision positive à l'intérieur de la classe et négative en dehors. Le modèle de détection des nouveautés repose sur le principe suivant : l'origine de l'espace est considérée comme étant la seule instance de la classe négative. Le problème revient, donc, à trouver un hyperplan qui sépare les observations de la classe cible de l'origine, et qui maximise la marge entre les deux [Scholkopf 2000].

Formellement, le problème est modélisé par le problème primal de programmation quadratique de l'équation [Hamel 2009] :

$$\begin{cases} \min\limits_{w,\xi,\rho} \frac{1}{2}\|w\|^2 + \frac{1}{\nu N}\sum\limits_{i=1}^{l}\xi_i - \rho \\ \langle w, \phi(\mathbf{x}_i)\rangle \geq \rho - \xi_i \\ \xi_i \geq 0 \quad i = 1, 2, \ldots, N \end{cases}$$

dont l'objectif est de maximiser la marge et de minimiser les erreurs de classification. La contrainte est la bonne classification des données d'apprentissage. N est le nombre d'observations de la classe cible, (w, ρ) les paramètres permettant de localiser l'hyperplan, ξ_i représentent les erreurs permises sur les exemples, pénalisées par le paramètre ν, et ϕ est une fonction de transformation d'espace dans ce modèle. Les données d'apprentissage $\mathbf{x}_i \in \mathcal{A}$ qui se situent du mauvais côté de l'hyperplan séparateur sont classées comme des nouveautés et la distance entre une nouveauté et l'hyperplan vaut $\frac{\xi_i}{\|w\|}$ avec $\xi_i > 0$. La distance entre la marge et l'origine est égale à $\frac{\rho}{\|w\|}$ et le paramètre $\nu \in [0, 1]$ est une borne supérieure du taux de nouveauté, mais aussi une borne inférieure du taux de vecteurs supports. Une fois (w, ρ) déterminés, toute nouvelle donnée pourra être classée par la fonction de décision de l'équation 2.16 :

$$f(\mathbf{x}) = \langle w, \phi(\mathbf{x}_i)\rangle - \rho \tag{2.16}$$

$\mathbf{x}$ appartient à la classe cible si $f(\mathbf{x})$ est positive. En fait, la résolution du problème primal de programmation quadratique formulé précédemment est réalisée par l'introduction des multiplicateurs de Lagrange pour obtenir le problème dual du système suivant :

$$\begin{cases} minimiser_{\alpha} & \frac{1}{2}\sum\limits_{i,j}\alpha_i\alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{sous contraintes} & \\ & \sum\limits_{i=1}^{n}\alpha_i = 1 \\ & 0 \leq \alpha_i \leq \frac{1}{\nu N} \end{cases}$$

Où K est un noyau qui représente la transformation d'espace ϕ. La fonction de décision pour toutes les données est la suivante :

$$f(x) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}) - \rho$$

Où ρ peut être déterminé à partir d'une donnée $\mathbf{x}_i$ d'apprentissage dont $\alpha_i = 0$ par l'équation suivante :

$$\rho = \sum_{j} \alpha_j K(\mathbf{x}_j, \mathbf{x}_i)$$

En résumé, on peut donc distinguer trois types d'observations :

- Les **nouveautés** pour lesquelles on a $f(x_i) < 0$ $\alpha_i = \frac{1}{\nu N}$ $\xi > 0$: correspondent aux données qui se situent au-dessus de l'hyperplan ;
- Les **observations normales** pour lesquelles on a $f(x_i) > 0$ $\alpha_i = \frac{1}{\nu N}$ $\xi = 0$ correspondent aux données qui se situent à l'intérieur de la marge (au-dessous de l'hyperplan) ;
- Les **vecteurs de support** pour lesquels on a $f(x_i) = 0$ $0 < \alpha_i < \frac{1}{\nu N}$ correspondent aux données qui se situent sur l'hyperplan.

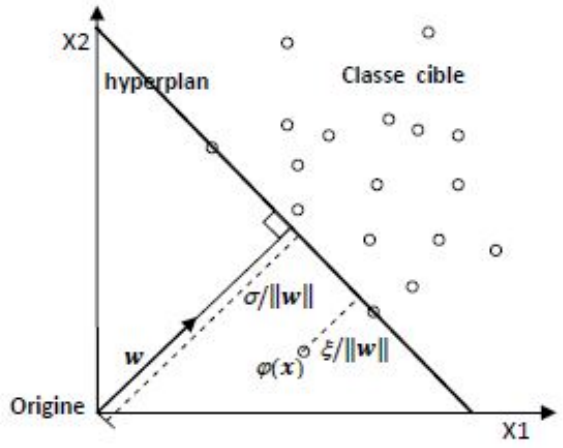


FIGURE 2.6 – Détection de nouveautés basée sur One-class SVM.

2.4.4 Autres méthodes de détection des nouveautés

La détection des nouveautés est l'une des exigences fondamentales d'une bonne classification, car parfois, les données de tests contiennent des informations qui n'étaient pas connues au moment de l'apprentissage du modèle.

Les réseaux de neurones sont aussi utilisés dans la détection des nouveautés [Markou 2003b]. Plusieurs types de réseaux de neurones peuvent traiter le problème, notamment le perceptron multi-couches [Moya 1993, Rusiecki 2012], les cartes auto-organisatrices [Ypma 1997, Xing 2009], les réseaux de Hopfield [Jagota 1991, Chandola 2009], les réseaux d'oscillation [Borisyuk 2004b], la reconnaissance des caractères [Singh 2006], etc. Une méthode de détection des nouveautés est généralement évaluée en utilisant le taux des vrais positifs et l'aire sous la courbe ROC (Receiver Operating Characteristics) [Markou 2003b]. Ces mesures sont utilisées pour l'évaluation des performances d'un classifieur.

2.5 Conclusion

Nous avons présenté dans ce second chapitre de cette thèse un panorama sur les différentes approches de la détection d'outliers et des nouveautés dans la littérature en introduisant le contexte de la classification non supervisée. La détection des groupes-outliers dans le cadre de l'apprentissage non supervisé est un problème intéressant et complexe. C'est ainsi que nous allons définir dans le chapitre suivant (chapitre 3) un nouveau modèle qui permet de détecter à la fois les groupes-outliers et les nouveautés.

Chapitre 3

Contribution : détection de groupes-outliers et des nouveautés en classification non supervisée

Sommaire

Dans cette première partie de nos travaux, nous introduisons une nouvelle mesure pour qualifier "l'outlier-ness" de chaque groupe/cluster. Cette mesure est intégrée et estimée dans un processus d'apprentissage non supervisé. Nous l'appelons par la suite "GOF" (Group Outlier Factor). Pour la validation, nous avons choisi d'intégrer cette mesure aux cartes topologiques [1]. Ceci permet d'apprendre la structure des données tout en fournissant un nouveau paramètre GOF. Ce paramètre est basé sur la densité et quantifie ainsi la particularité du groupe (cluster) : plus la valeur est grande, plus le groupe est susceptible d'être un groupe outlier.

1. Voir le chapitre 1 section 1.3.4

3.1 Cartes auto-organisatrices et référents outliers

Soit $\mathcal{D}$ l'ensemble des données $\mathbf{x}_i$ d'apprentissage, de taille N, où chaque observation $\mathbf{x}_i = (x_i^1, x_i^2, ..., x_i^j, ..., x_i^d) \in \Re^d$. Notre approche propose un apprentissage à l'aide des cartes topologiques tout en détectant les groupes-outliers. Nous rappelons qu'un groupe-outlier n'est pas nécessairement un groupe aberrant ; il peut être un groupe d'intérêt, de nouveauté, etc. En fait, c'est un groupe qui a un comportement largement différent du reste des données. Ce type de groupe peut biaiser les résultats comme il peut constituer un échantillon exhaustif.

Le modèle classique des cartes auto-organisatrices se présente sous forme d'une grille possédant un ordre topologique de K cellules. Les cellules sont réparties sur les nœuds d'un maillage. La prise en compte dans la carte de la notion de proximité impose de définir une relation de voisinage topologique. L'influence mutuelle entre deux cellules c et r est donc définie par la fonction $\mathcal{K}^{\mathcal{T}}(\delta(c,r))$ où $\delta(c,r)$ est la distance de graphe entre les deux cellules c et r. Dans notre approche, chaque cellule c de la grille $\mathcal{C}$ est associée à la fois à deux paramètres : un vecteur référent $\mathbf{w}_c = (w_c^1, w_c^2..., w_c^j, ..., w_c^d)$ de dimension d et une nouvelle valeur que nous proposons d'appeler "GOF" (Group Outlier Factor). On note par la suite $\mathcal{W} = \{\mathbf{w}_c, \mathbf{w}_c \in \Re^d\}_{c=1}^K$ l'ensemble des référents et par $GOF_c \in \Re$ l'indicateur "d'outlier-ness" associé à chaque cellule c.

Chaque référent est associé à un sous-ensemble de données affectées à la cellule c qui sera noté P_c. L'ensemble des sous ensembles forme la partition de l'ensemble des données $\mathcal{D}$, $\mathcal{P} = \{P_1, ...P_c, ..., P_K\}$ où $P_c = \{\mathbf{x}_i, \phi(\mathbf{x}_i) = c\}$.

Nous rappelons ici que la fonction d'affectation ϕ est définie de la manière suivante :

$$\phi(\mathbf{x}_i) = \arg \min_{1 \leq c \leq K} \parallel \mathbf{x}_i - \mathbf{w}_c \parallel^2$$

Chaque cellule c est associée à une valeur réelle GOF_c qui indique "l'outlier-ness" de la cellule et qui résume en d'autre terme "l'outlier-ness" de toutes les observations $\mathbf{x}_i$ affectées à la cellule c. De la même manière que l'approche LOF définie dans l'algorithme 7, l'estimation de "l'outlier-ness" de chaque observation est liée à la densité. Chaque cellule de la carte est associé à un référent. La fonction $f_c(x)$ permet d'estimer la densité des données au niveau de chaque cellule c définie comme suit :

$$f_c(\mathbf{x}_i) = \exp^{-} \frac{\parallel \mathbf{x}_i - \mathbf{w}_c \parallel^2}{2\sigma^2}$$

où le paramètre σ est l'écart type standard entre les données. Le choix du paramètre σ est important et sa valeur optimale est difficile à calculer. En effet,

si σ est trop grand, alors la répartition des données va influencer les valeurs de densité de tous les prototypes, les prototypes proches sont alors associés à des densités similaires, ce qui induit une diminution de la précision de l'estimation. Cependant, si σ est trop petit, une grande proportion des données (les plus éloignées des prototypes) n'influenceront pas les valeurs de la densité des prototypes, ce qui induit une perte d'information.

Dans notre cas, la densité est définie par une fonction de type gaussienne. Ainsi, par analogie à l'approche LOF présentée dans le chapitre 2, nous proposons d'estimer "l'outlier-ness" de chaque observation $\mathbf{x}$ associé à un référent $\mathbf{w}_c$ en utilisant l'expression suivante :

$$OF_c(\mathbf{x}_i) = \frac{\frac{\sum_{\mathbf{x}_j \in P_c} \frac{1}{f_c(\mathbf{x}_j)}}{|P_c|}}{\frac{1}{f_c(\mathbf{x}_i)}}$$

Dans le cas particulier des cartes topologiques, nous proposons de minimiser la fonction de coût suivante :

$$\mathcal{R}(\mathcal{W}, GOF, \phi) = \mathcal{R}(\mathcal{W}, \phi) + \mathcal{R}(GOF)$$

où

$$\mathcal{R}(\mathcal{W}, \phi) = \sum_{i=1}^{N} \sum_{c=1}^{K} \mathcal{K}^{\mathcal{T}}(\delta(\phi(\mathbf{x}_i), c)) \|\mathbf{w}_c - \mathbf{x}_i\|^2$$

et

$$\begin{aligned}\mathcal{R}(GOF) &= \sum_{i=1}^{N} \sum_{c=1}^{K} \mathcal{K}^{\mathcal{T}}(\delta(\phi(\mathbf{x}_i), c)) \left(GOF_c - OF_c(\mathbf{x}_i)\right)^2 \\ &= \sum_{i=1}^{N} \sum_{c=1}^{K} \mathcal{K}^{\mathcal{T}}(\delta(\phi(\mathbf{x}_i), c)) \left(GOF_c - \frac{\frac{\sum_{\mathbf{x}_j \in P_c} \frac{1}{f_c(\mathbf{x}_j)}}{|P_c|}}{\frac{1}{f_c(\mathbf{x}_i)}} \right)^2\end{aligned}$$

La notion de voisinage est introduite par la fonction noyau :

$$\mathcal{K}^T(\delta(c, r)) = \exp\left(\frac{-\delta(c, r)}{T}\right)$$

Où T est la température qui varie de T_{max} à T_{min}.

Le premier terme $\mathcal{R}(\mathcal{W}, \phi)$ dépend des paramètres $\mathcal{W}$ et permet d'estimer les référents. Le deuxième terme est le coût $\mathcal{R}(GOF)$ lié à l'estimation des

valeurs GOF associées à chaque cellule. L'algorithme d'apprentissage suivant (algorithme 9) propose une solution pour la minimisation de la fonction coût en utilisant la méthode de la descente du gradient.

Chaque paramètre sera "récompensé" par une augmentation de sa valeur. Cette valeur est d'autant plus importante que l'apprentissage est avancé et que les référents représentent bien les données. Cependant, les autres seront "punies" par une diminution de leurs valeurs. Ainsi, à la fin de l'apprentissage, un ensemble de prototypes $\mathbf{w}_c$ et de score GOF_c sera représentatif d'un sous-groupe P_c ou cluster de l'ensemble des données.

Algorithme 9 : Algorithme GOF-SOM

1: **Entrées :**
   - Les données $\mathcal{D} = \{\mathbf{x}_i\}_{i=1..N}$
   - La carte SOM avec K référents initialisés $\{\mathbf{w}_c, c = 1 \dots K\}$
   - t_{max} : le nombre maximum d'itérations.
   - Initialisation des valeurs GOF.

2: **Sorties :**
   - Une partition $P = \{P_c\}_{c=1...K}$.
   - Les valeurs de GOF= $\{GOF_c, c = 1 \dots K\}$

3: **Phase de compétition** : affecter une donnée $\mathbf{x}_i$ en utilisant la fonction

$$\phi(\mathbf{x}_i) = \arg \min_{1 \leq c \leq K} \parallel \mathbf{x}_i - \mathbf{w}_c \parallel^2$$

4: **Phase d'adaptation**
   - Mettre à jour les référents $\mathbf{w}_c$ de chaque cellule c

$$\mathbf{w}_c(t) = \mathbf{w}_c(t-1) - \varepsilon(t)\mathcal{K}^{\mathcal{T}}(\delta(\phi(\mathbf{x}_i), c)))\,(\mathbf{w}_c(t-1) - \mathbf{x}_i)$$

   - Mettre à jour les valeurs de GOF_c associées à chaque cellule c

$$GOF_c(t) = GOF_c(t-1) - \varepsilon(t)\mathcal{K}^{\mathcal{T}}(\delta(\phi(\mathbf{x}_i), c))) \left(GOF_c(t-1) - \frac{\frac{\sum_{\mathbf{x}_j \in P_c} \frac{1}{f_c(\mathbf{x}_j)}}{|P_c|}}{\frac{1}{f_c(\mathbf{x}_i)}} \right)$$

     où $\varepsilon(t)$ est le pas d'apprentissage, T est la température qui varie au cours de l'apprentissage.

5: Répéter les phases de compétition et d'adaptation jusqu'à un nombre d'itérations fixé $t = t_{max}$.

Cet algorithme est proche de l'algorithme classique SOM, ce qui permet de conserver la topologie en deux dimensions de la carte, et de fournir une visua-

lisation simple de la structure des données. Par ailleurs, l'utilisation d'une valeur GOF_c donne une information locale sur "l'outliner-ness" du sous-ensemble associé à la cellule c, et garde une information sur la structure générale des données et des clusters entre eux.

Il est à noter que le résultat final dépend en partie de l'ordre de présentation des données (cet ordre est souvent aléatoire) et peut donc varier légèrement d'une exécution à l'autre. Les résultats dépendent aussi de l'initialisation des référents de la carte (qui peut être aléatoire). Concernant l'initialisation des GOF_c, nous avons opté pour une initialisation équiprobable. Nous avons évalué les performances de cet algorithme sur un ensemble de jeux de données présentant des difficultés pour la classification.

3.2 Détection des nouveautés

3.2.1 Classifieur "GOF-Novelty"

Dans cette section, nous allons montrer comment utiliser la mesure GOF pour la détection des nouveautés. Nous utilisons la même structure et architecture des cartes déjà définies dans la section 5.4. Nous proposons un classifieur pour la détection des nouveautés basée sur le GOF associé à la nouvelle base $\mathcal{D}^{'}$. La méthode que nous proposons consiste à affecter les données $\mathbf{x}_i$ en utilisant les résultats de l'algorithme GOF-SOM. Aussi, nous proposons de calculer pour chaque cluster c et pour chaque donnée $\mathbf{x}_i$ un score "Outlier Factor" ($OF_c(\mathbf{x}_i)$) comme suit :

$$OF_c(\mathbf{x}_i) = \frac{\frac{\sum_{\mathbf{x}_j \in P_c} \frac{1}{fc(\mathbf{x}_j)}}{|P_c|}}{\frac{1}{f_c(\mathbf{x}_i)}}$$

Si la valeur de $OF_c(\mathbf{x}_i)$ est plus grande que la valeur de Group Outlier Factor du cluster (GOF_c), alors, nécessairement, la donnée $\mathbf{x}_i$ est nouvelle. C'est ainsi que nous proposons l'algorithme 10 qui permet de traiter le problème de la détection des nouveautés en utilisant le paramètre GOF.

Algorithme 10 Algorithm GOF-Novelty

Entrées :
- La partition $P = \{P_c\}_{c=1..K}$,
- Valeurs de GOF = $\{GOF_c, c = 1..K\}$,
- La nouvelle base de données $\mathcal{D}^{'} = \{\mathbf{x}_i\}_{i=1..M}$.

Sorties :
- $(Novelty\_label)$: vecteur binaire des nouveautés.

pour i=1 : M **faire**

$OF_{\phi(\mathbf{x}_i)}(\mathbf{x}_i) = \frac{\frac{\sum_{\mathbf{x}_j \in P_{\phi(\mathbf{x}_i)}} \frac{1}{fc(\mathbf{x}_j)}}{|P_{\phi(\mathbf{x}_i)}|}}{\frac{1}{f_c(\mathbf{x}_i)}}$

$Dif = |OF_{\phi(\mathbf{x}_i)}(\mathbf{x}_i) - GOF_{\phi(\mathbf{x}_i)}|$

si $(Dif < threshold)$ **alors**

$Novelty\_label(\mathbf{x}_i) = 0$;

sinon

$Novelty\_label(\mathbf{x}_i) = 1$;

Threshold peut varier selon les bases (σ par défaut).

3.3 Expérimentations et évaluations du score GOF

Nous avons utilisé des bases de données provenant du répertoire UCI [Frank 2010] ainsi que des bases simulées, qui vont nous permettre de valider notre approche. Le tableau 3.1 présente la description des bases “simulées“ et des bases ”publiques”.

Nom de la base	Nombre de données	Taille de la carte
base simulée 1	160	5×13
base simulée 2	234	3×26
base simulée 3	569	8×15
base simulée 4	402	8×13
anneauxModif	1072	14×12
demicercleModif	638	13×10
HeptaModif	212	9×8
LsunModif	400	11×9
TargetModif	951	13×12
GolfBallModif	4343	19×17

TABLE 3.1 – Description des bases jouées et publiques.

Remarques :

- La taille de la carte est choisie selon l'heuristique de Kohonen [2] $Taille = 5 \times K^{0.54321}$. Dans certain cas, nous choisissons une taille proportionnelle à la taille de la base de données ;
- L'initialisation des prototypes est réalisé d'une façon aléatoire ;
- L'initialisation des valeurs de GOF est réalisée d'une manière équiprobable $= \frac{1}{K}$;
- Les bases simulées sont générées aléatoirement suivant une loi normale (une gaussienne) pour créer des groupes largement isolés du reste des données.

3.3.1 Mesures de performances

Afin de calculer les critères de performance de l'algorithme GOF-Novelty, nous avons défini d'abord la matrice de confusion [Kohavi 1998] (voir tableau 3.2), qui contient les informations sur les classes réelles et les classes prédites. Ensuite, nous calculons les indices : rappel, précision, f-mesure et AUC, afin d'évaluer les performances de l'algorithme GOF-Novelty.

2. http ://www.cis.hut.fi/somtoolbox/package/docs2/som_make.html

		Classes réelles +	**Classes réelles** −
Classes prédites	+	Vrais Positives (VP)	Faux Positives (FP)
	−	Faux Negatives (FN)	Vrais Negatives (VN)

TABLE 3.2 – Matrice de confusion.

Les formules de calcul des indices du rappel, précision, F-mesure et AUC sont définies comme suit ([Powers 2007]) :

– Le rappel est le pourcentage des données positives bien classées. Cet indice est aussi appelé : sensibilité, taux de vrais positifs (TVP) ou recall.

$$Rappel = \frac{VP}{VP + FN}$$

– L'indice de précision est la proportion de données prédictives positives correctement classées :

$$Precision = \frac{VP}{VP + FP}$$

– La F-mesure est une combinaison pondérée de rappel et précision :

$$F - mesure = \frac{2 \times rappel \times precision}{precision + rappel}$$

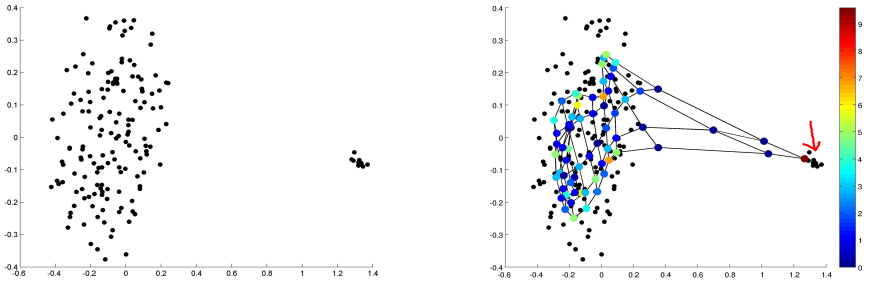
– La courbe Receiver Operating Characteristic (ROC) est un graphique exprimant la capacité d'un classifieur à faire la distinction entre les classes positives et les classes négatives. L'aire sous la courbe ROC (AUC) indique la précision d'un classifieur.

$$AUC = \frac{TVP + TFN}{2}$$

3.3.2 Résultats visuels

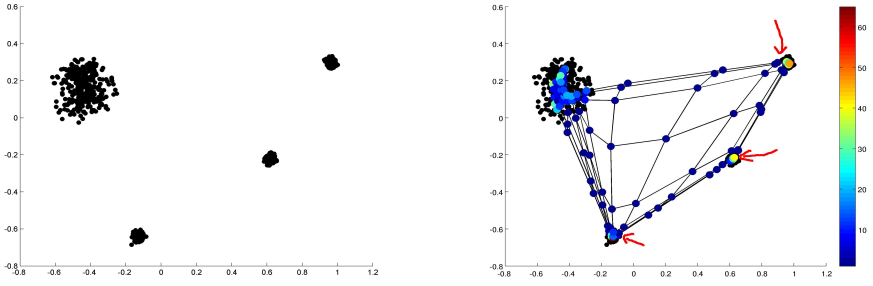
Afin de vérifier visuellement les résultats, nous avons visualisé les données avec les référents de la carte. La nouvelle information fournie dans ces figures est la visualisation d'un score "d'outlier-ness" de chaque groupe de données.

Ce score nommé “GOF”, estimé au cours de l’apprentissage, est visualisé à l’aide d’une couleur associée à chaque cellule de la carte. Plus la couleur est rouge, plus le groupe a une forte valeur de GOF.



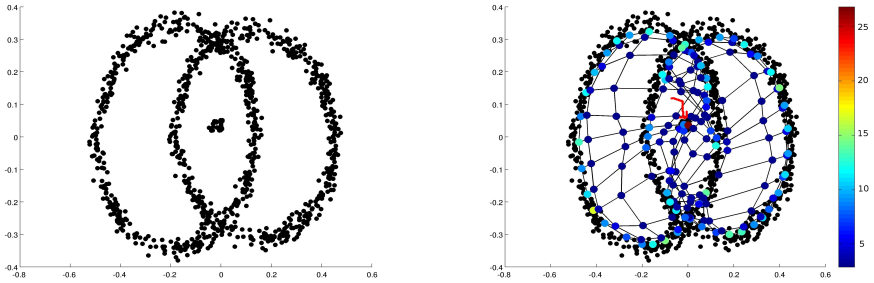
(a) Données de la base simulée 1 (b) Carte associée aux données de la base simulée 1

FIGURE 3.1 – GOF-SOM appliqué sur les données de la base simulée 1



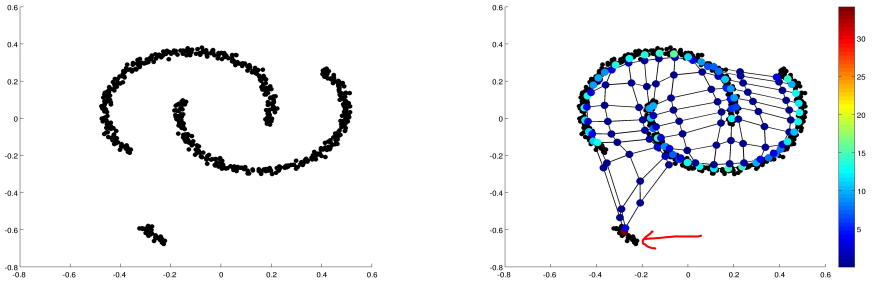
(a) Données de la base simulée 3 (b) Carte associée aux données de la base simulée 3

FIGURE 3.2 – GOF-SOM appliqué sur les données de la base simulée 3



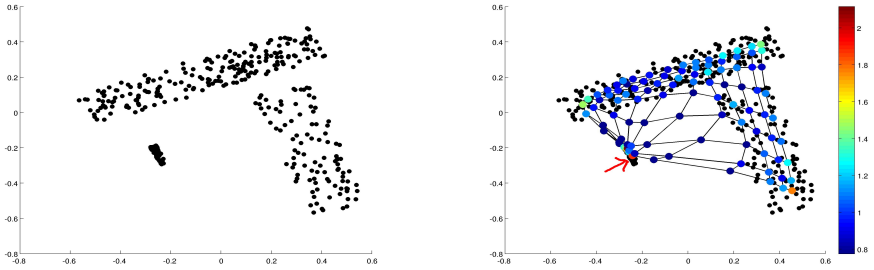
(a) Données de la base anneauxModif (b) Carte associée aux données de la base anneauxModif

FIGURE 3.3 – GOF-SOM appliqué sur les données de la base anneauxModif



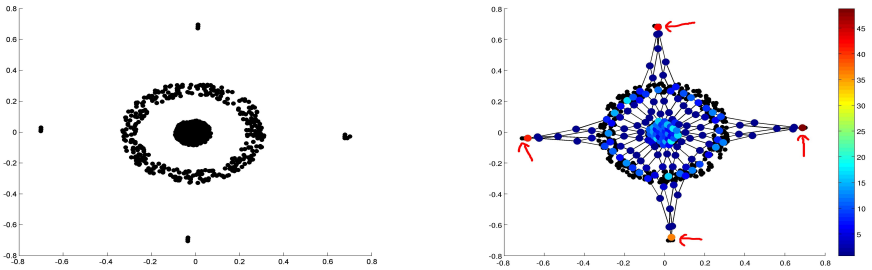
(a) Données de la base demicercleModif (b) Carte associée aux données de la base demicercleModif

FIGURE 3.4 – GOF-SOM appliqué sur les données de la base demicercleModif



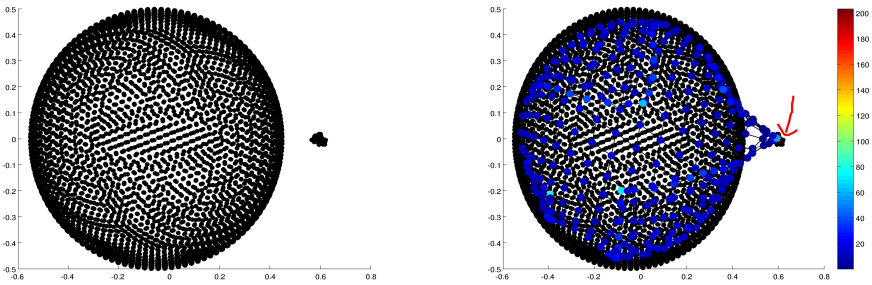
(a) Données de la base LsunModif (b) Carte associée aux données de la base HeptaModif

FIGURE 3.5 – GOF-SOM appliqué sur les données de la base LsunModif



(a) Données de la base TargetModif (b) Carte associée aux données de la base TargetModif

FIGURE 3.6 – GOF-SOM appliqué sur les données de la base TargetModif



(a) Données de la base GolfBallModif (b) Carte associée aux données de la base GolfBallModif

FIGURE 3.7 – GOF-SOM appliqué sur les données de la base GolfBallModif

Les figures 3.1 à 3.8 représentent les données (figure de gauche) et la projection de la carte avec la valeur GOF estimée (figure de droite). Nous observons que les groupes-outliers sont clairement visibles et sont associés à des valeurs GOF très fortes représentées par la couleur rouge (colonne de droite). Les référents et le paramètre GOF s'adaptent parfaitement et simultanément avec les groupes isolés.

3.3.3 Critère de sélection des groupes-outliers : "*Scree Acceleration Test*"

Afin de détecter les groupes-outliers, nous avons utilisé un test statistique proposé par [Cattell 1966] appelé "Scree Test". Ce test nous permet de faire

une sélection des valeurs GOF d'une manière automatique. L'utilisation initiale du test "Scree Test" [Cattell 1966] consiste en la détermination visuelle du nombre de valeurs propres à prendre en compte lors d'une analyse en composantes principales. L'idée de base est de représenter graphiquement les valeurs propres et de trouver la valeur pour laquelle le changement brutal est émergé (scree). Le nombre de composantes à garder correspond au nombre de valeurs propres précédant ce "Scree". Fréquemment, ce "Scree" apparaît là où la pente du graphe change radicalement. Ainsi, il s'agit de trouver la décélération maximale dans ce graphique.

L'utilisation de ce test sur notre vecteur de paramètre GOF consiste à détecter, par exemple, le changement brutal dans le vecteur

$$GOF = (GOF_1, GOF_2..., GOF_i, ..., GOF_K)$$

Ainsi, il faudrait détecter la plus forte décélération. La procédure de sélection est composée des étapes suivantes (voir algorithme 11) :

Algorithme 11 : Algorithme scree test adapté à GOF

1: Ordonner le vecteur $GOF = (GOF_1, GOF_2..., GOF_i, ..., GOF_K)$ suivant un ordre décroissant. Le nouveau vecteur ordonné est noté $GOF = (GOF^1_{..}, GOF^2_{..}..., GOF^i_{..}, ..., GOF^K_{..})$ où l'exposant i de GOF^i indique l'ordre.
2: Calculer les premières différences $df_i = GOF^i_{..} - GOF^{i+1}_{..}$
3: Calculer les deuxièmes différences (l'accélération) $acc_i = df_i - df_{i+1}$
4: Chercher le changement brutal "scree" à l'aide de la fonction suivante : $\max_i (|acc_i| + |acc_{i+1}|)$
Ce processus permet de sélectionner toutes les composantes se trouvant avant le changement brutal.

Nous souhaitons sélectionner d'une manière automatique les groupes-outliers. À cet effet, nous avons utilisé le test statistique "Scree Test". Le tableau 3.4 présente les résultats obtenus après l'application du "Scree Test".

Base de données	# G.O réels	# G.O "Scree Test"	# G.O sans répétition
anneauxModif	1	1	1
demicerModif	1	1	1
HeptaModif	1	1	1
LsunModif	1	1	1
TargetModif	4	4	4
GolfBallModif	1	1	1
base simulée 1	1	1	1
base simulée 2	2	2	2
base simulée 3	3	5	3
base simulée 4	4	6	4

TABLE 3.3 – Détection automatique des groupes-outliers.

Chaque valeur GOF sélectionnée représente un groupe-outlier. Il existe des cas où plusieurs référents sélectionnés décrivent ensemble le même cluster. Par exemple, dans le cas de la base simulée 3, "Scree Test" a sélectionné 5 groupes-outliers dont 2 groupes sont des sous-ensembles du cluster outlier simulé, les deux autres appartiennent à un autre cluster et le dernier groupe-outlier représente le 3e cluster. Finalement, les groupes-outliers sélectionnés ne détectent que 3 clusters.

Afin de montrer l'intérêt de détecter les groupes-outliers, nous avons calculé le paramètre LOF pour chacune des bases de données. Particulièrement pour les bases simulées, nous avons constaté que l'utilisation de LOF ne permet de détecter aucune donnée outlier. Cet inconvénient est connu, car LOF repose sur deux principes : la distance entre les données et la densité de chacune d'elles. Généralement, dans un cluster, les distances entre les données sont petites, et les densités locales des données, en les comparant avec les densités moyennes de leurs *k-ppv*, restent relativement égales. Ainsi, dans le cas de nos bases simulées, les valeurs du LOF seront presque les mêmes que celles des données normales.

3.4 Expérimentations et évaluations de la détection des nouveautés

Dans nos expérimentations, l'ensemble des données d'apprentissage est formé uniquement des données étiquetées 0 (pas de nouveautés). Les bases de test contiennent des données étiquetées 1 (nouvelles) et 20 % de données étiquetées 0. Le tableau 3.4 représente la description des bases de données utilisées dans ces experimentations. Les bases de données utilisées sont de type "One-class classifier" [Pekalska 2003] téléchargeables à partir de cette adresse : *http ://homepage.tudelft.nl/n9d04/occ/index.html*.

Nous utilisons aussi les bases de données exposées dans le tableau 3.1 et qui sont découpées exactement de la même manière que celles présentées dans le tableau 3.4. Pour ces bases, les étiquettes 1 (nouvelles) sont attribuées à la base minoritaire.

Base de données	# Observations	# Variables	# Normales	# Outliers
Iris Setosa	150	4	50	100
Iris Virginica	150	4	50	100
Sonar Mines	108	60	11	97
Biomed Healthy	194	5	127	67
Hepatitis Normal	155	19	123	32
Diabetes Present	768	8	500	268
Ecoli Periplasm	336	7	52	284
Spectf 1	349	44	254	95
Balance-Scale	625	4	288	337
Glass Building	214	9	70	144
Waveform 2	900	21	300	600

TABLE 3.4 – Description des bases de données utilisées pour la détection des nouveautés.

Nous cherchons à savoir si le classifieur que nous avons construit permet d'estimer d'une manière efficace les données nouvelles. C'est pour cela qu'un des principaux critères d'évaluation de la détection des nouveautés est le rappel ([Markou 2003b]). À cet effet, nous avons sélectionné ce critère (rappel) ainsi que trois autres critères d'évaluation des performances de notre algorithme, en l'occurrence, la précision, la F-measure et l'AUC. Afin de calculer ces critères, nous avons utilisé la matrice de confusion [Kohavi 1998] présentée dans le tableau 3.2.

3.4.1 Validation croisée

La validation croisée est une technique de ré-échantillonnage permettant d'estimer le taux d'erreur d'un classifieur. La procédure suivie est la suivante : nous divisons la base en 5 sous-ensembles. Pour chaque expérimentation, nous sélectionnons 4 sous-ensembles pour la base d'apprentissage et un sous-ensemble pour la base de teste. Ce processus est répété 10 fois. Nous avons comparé GOF-Novelty avec l'approche ACP ([Hoffmann 2007]) en prenant en compte 2 composantes principales et le One-SVM ([Scholkopf 2001]).

Les tableaux 3.5, 3.6, 3.7 et 3.8 représentent les résultats expérimentaux des indices du rappel, précision, F-mesure et l'aire sous la courbe (AUC).

Les mêmes résultats sont représentés dans les figures 3.8, 3.9, 3.10 et 3.11 sous forme de graphiques en radar. Nous rappelons que tous ces critères sont calculés sur la base de test.

Bases	GOF-Novelty		ACP		One-SVM	
	Moyenne	Ecart	Moyenne	Ecart	Moyenne	Ecart
b s 1	0.750	± 0.033	0.723	± 0.092	0.852	± 0.081
b s 2	0.508	± 0.067	0.522	± 0.121	0.843	± 0.059
b s 3	0.490	± 0.030	0.382	± 0.043	0.529	± 0.062
b s 4	0.453	± 0.046	0.412	± 0.051	0.521	± 0.051
demicercleM	0.625	± 0.100	0.213	± 0.038	0.272	± 0.025
anneauxM	0.800	± 0.001	0.713	± 0.018	0.672	± 0.021
LsunM	0.722	± 0.056	0.798	± 0.039	0.213	± 0.043
TargetM	0.709	± 0.016	0.715	± 0.028	0.590	± 0.002
HeptaM	0.586	± 0.066	0.512	± 0.042	0.240	± 0.051
GolfBallM	0.836	± 0.061	0.703	± 0.041	0.119	± 0.032
Iris S	0.698	± 0.073	0.431	± 0.081	0.762	± 0.061
Sonar M	0.673	± 0.092	0.381	± 0.058	0.352	± 0.031
Biomed H	0.520	± 0.092	0.721	± 0.081	0.393	± 0.063
Hepatitis N	0.689	± 0.056	0.619	± 0.045	0.538	± 0.131
Diabetes P	0.666	± 0.072	0.532	± 0.062	0.712	± 0.059
Ecoli P	0.980	± 0.001	0.818	± 0.013	0.883	± 0.009
Spectf 1	0.620	± 0.060	0.723	± 0.082	0.731	± 0.091
Balance S L	0.783	± 0.013	0.853	± 0.042	0.520	± 0.031
Glass B F	0.584	± 0.022	0.601	± 0.058	0.661	± 0.018
Waveform 2	0.823	± 0.091	0.539	± 0.083	0.453	± 0.031

TABLE 3.5 – Moyenne et écart de l'indice du rappel obtenus sur GOF-Novelty, ACP et OneSVM en utilisant une validation croisée. bs : base simulée

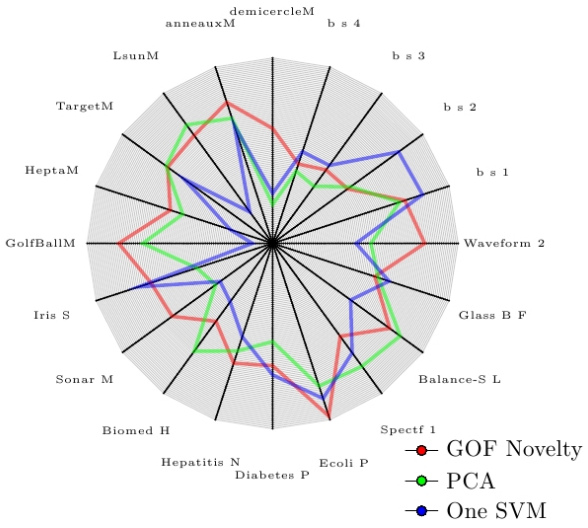


FIGURE 3.8 – L'indice du rappel en utilisant une validation croisée représenté sous forme d'un radar.

Bases	GOF-Novelty		ACP		One-SVM	
	Moyenne	Ecart	Moyenne	Ecart	Moyenne	Ecart
b s 1	0.343	± 0.067	0.521	± 0.053	0.431	± 0.012
b s 2	0.483	± 0.082	0.631	± 0.092	0.420	± 0.061
b s 3	0.842	± 0.028	0.591	± 0.043	0.723	± 0.031
b s 4	0.700	± 0.047	0.721	± 0.040	0.713	± 0.038
demicercleM	0.498	± 0.024	0.513	± 0.032	0.621	± 0.059
anneauxM	0.527	± 0.005	0.502	± 0.012	0.343	± 0.009
LsunM	0.771	± 0.036	0.629	± 0.029	0.812	± 0.048
TargetM	0.961	± 0.012	0.912	± 0.009	0.953	± 0.014
HeptaM	0.830	± 0.030	0.892	± 0.041	0.431	± 0.021
GolfBallM	0.753	± 0.009	0.432	± 0.093	0.620	± 0.021
Iris S	0.637	± 0.101	0.652	± 0.098	0.628	± 0.063
Sonar M	0.457	± 0.016	0.562	± 0.142	0.493	± 0.041
Biomed H	0.348	± 0.016	0.432	± 0.023	0.160	± 0.043
Hepatitis N	0.753	± 0.041	0.812	± 0.058	0.661	± 0.023
Diabetes P	0.698	± 0.018	0.654	± 0.028	0.682	± 0.034
Ecoli P	0.812	± 0.082	0.445	± 0.030	0.809	± 0.068
Spectf 1	0.286	± 0.054	0.352	± 0.110	0.221	± 0.031
Balance S L	0.832	± 0.008	0.453	± 0.049	0.738	± 0.020
Glass B F	0.659	± 0.035	0.693	± 0.032	0.662	± 0.030
Waveform 2	0.738	± 0.020	0.712	± 0.018	0.703	± 0.024

TABLE 3.6 – Moyenne et écart de l'indice de précision obtenus sur GOF-Novelty, ACP et One-SVM en utilisant une validation croisée. bs : base simulée

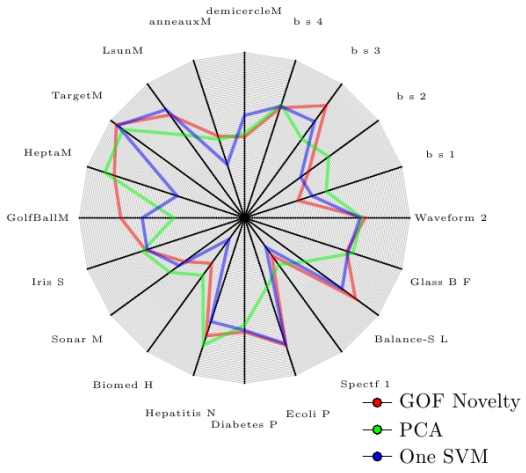


FIGURE 3.9 – L'indice de la précision en utilisant une validation croisée représenté sous forme d'un radar.

Bases	GOF-Novelty		ACP		One-SVM	
	Moyenne	Ecart	Moyenne	Ecart	Moyenne	Ecart
b s 1	0.471	± 0.044	0.606	± 0.067	0.572	± 0.021
b s 2	0.495	± 0.074	0.571	± 0.105	0.561	± 0.060
b s 3	0.619	± 0.029	0.464	± 0.043	0.611	± 0.041
b s 4	0.550	± 0.046	0.524	± 0.045	0.602	± 0.044
demicercleM	0.554	± 0.038	0.301	± 0.035	0.378	± 0.035
anneauxM	0.635	± 0.002	0.589	± 0.014	0.454	± 0.013
LsunM	0.746	± 0.044	0.703	± 0.033	0.337	± 0.045
TargetM	0.816	± 0.014	0.802	± 0.014	0.729	± 0.004
HeptaM	0.687	± 0.041	0.651	± 0.041	0.308	± 0.030
GolfBallM	0.792	± 0.016	0.535	± 0.057	0.200	± 0.025
Iris S	0.666	± 0.085	0.519	± 0.089	0.689	± 0.062
Sonar M	0.544	± 0.027	0.454	± 0.082	0.411	± 0.035
Biomed H	0.417	± 0.027	0.540	± 0.015	0.227	± 0.020
Hepatitis N	0.720	± 0.047	0.702	± 0.051	0.593	± 0.039
Diabetes P	0.682	± 0.029	0.587	± 0.039	0.697	± 0.043
Ecoli P	0.888	± 0.002	0.576	± 0.018	0.844	± 0.016
Spectf 1	0.391	± 0.057	0.473	± 0.094	0.339	± 0.046
Balance S L	0.807	± 0.010	0.592	± 0.045	0.610	± 0.024
Glass B F	0.619	± 0.027	0.644	± 0.041	0.661	± 0.022
Waveform 2	0.778	± 0.033	0.614	± 0.030	0.551	± 0.027

TABLE 3.7 – Moyenne et écart de l'indice de la F-mesure obtenus sur GOF-Novelty, ACP et One-SVM en utilisant une validation croisée. bs : base simulée

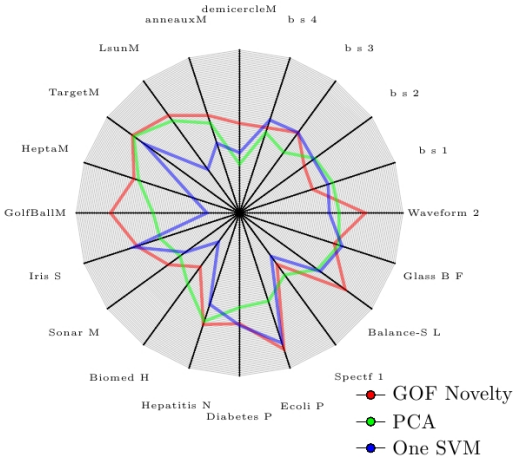


FIGURE 3.10 – L'indice de la F-mesure en utilisant une validation croisée représenté sous forme d'un radar.

Bases	GOF-Novelty		ACP		One-SVM	
	Moyenne	Ecart	Moyenne	Ecart	Moyenne	Ecart
b s 2	0.505	± 0.062	0.662	± 0.072	0.482	± 0.068
b s 3	0.528	± 0.055	0.432	± 0.040	0.503	± 0.012
b s 4	0.528	± 0.039	0.500	± 0.035	0.430	± 0.020
demicercleM	0.499	± 0.030	0.431	± 0.026	0.381	± 0.031
anneauxM	0.488	± 0.018	0.610	± 0.052	0.512	± 0.021
LsunM	0.537	± 0.071	0.380	± 0.063	0.483	± 0.059
TargetM	0.556	± 0.086	0.681	± 0.162	0.502	± 0.061
HeptaM	0.439	± 0.093	0.503	± 0.082	0.501	± 0.129
GolfBallM	0.693	± 0.039	0.439	± 0.021	0.538	± 0.123
Iris S	0.484	± 0.064	0.531	± 0.092	0.582	± 0.072
Sonar M	0.488	± 0.014	0.491	± 0.018	0.494	± 0.021
Biomed H	0.494	± 0.019	0.397	± 0.020	0.502	± 0.027
Hepatitis N	0.638	± 0.019	0.603	± 0.016	0.651	± 0.031
Diabetes P	0.830	± 0.018	0.431	± 0.031	0.520	± 0.002
Ecoli P	0.793	± 0.093	0.821	± 0.089	0.753	± 0.037
Spectf 1	0.496	± 0.027	0.382	± 0.031	0.431	± 0.030
Balance S l	0.703	± 0.006	0.802	± 0.020	0.721	± 0.012
Glass B L	0.513	± 0.028	0.472	± 0.011	0.500	± 0.021
Waveform 2	0.712	± 0.025	0.698	± 0.019	0.621	± 0.038

TABLE 3.8 – Moyenne et écart de l'indice de l'AUC obtenus sur GOF-Novelty, ACP et One-SVM en utilisant une validation croisée. bs : base simulée

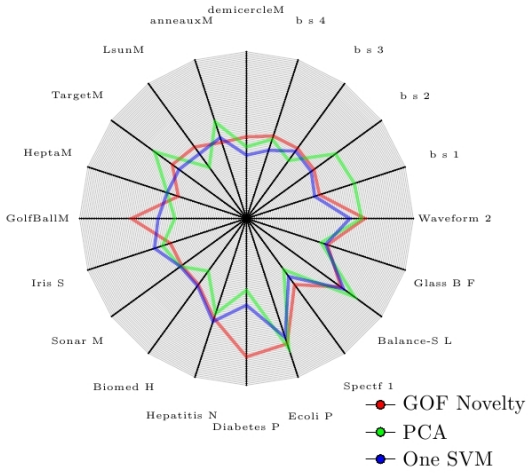


FIGURE 3.11 – L'indice de l'AUC en utilisant une validation croisée représenté sous forme d'un radar.

Indice du rappel : le tableau 3.5 résume les résultats expérimentaux obtenus sur l'indice du rappel. Notre méthode GOF-Novelty fournit les valeurs les plus élevées de l'indice du rappel pour les bases de données suivantes : demicercleM, anneauxM, HeptaM, GolfBallM, Sonar Mines, Hepatitis Normal, Ecoli Periplasm et Waveform 2.

One-SVM donne de meilleurs résultats dans les bases : simulées 1, 2, 3 et 4, Iris Setosa, Spectf 1 et Glass Building Float. L'ACP est plus performante dans les bases LsunM, TargetM, Biomed Healthy et Balance-Scale Left.

En dépit d'une diminution des performances dans les bases précédentes, GOF-Novelty reste la méthode la plus stable en comparaison avec les approches ACP et One-SVM. Par exemple, dans le jeu de données demicercleM, GOF-Novelty fournit 0.625. Nous observons clairement la baisse de l'indice du rappel dans les approches ACP et One-SVM (0.213 et 0.272 respectivement).

Indice de précision : l'analyse de l'indice de précision présenté dans le tableau 3.6 montre que GOF-Novelty, ACP et One-SVM fournissent des valeurs équivalentes dans les bases de données suivantes : Iris Setosa, Sonar Mines, Hepatitis Normal, Diabets Present, Spectf 1, Glass-Building Float, Waveform 2. Une diminution de performance pour notre approche GOF-Novelty est observée dans les bases simulées 1 et 2 et Biomed Healthy. GOF-Novelty fournit les valeurs les plus élevées de l'indice de précision dans les bases : simulée 3, GolfBall, Ecoli Periplasm et Balance-Scale left.

On observe une faible diminution en termes d'indice de précision dans les

approches ACP et One-SVM dans certaines bases de données. Par exemple, dans les bases Hepta et Biomed Healthy, One-SVM fournit respectivement 0.431 et 0.16 alors que la meilleure valeur de l'indice de précision est fournie par l'approche ACP (0.892 et 0.432 respectivement). Notre méthode GOF-Novelty reste compétitive à l'approche ACP, où elle fournit 0.83 et 0.348 respectivement.

Indice de F-mesure : observant l'indice F-mesure résumé dans le tableau 3.7, notre méthode GOF-Novelty fournit les valeurs les plus élevées de la F-mesure dans plusieurs bases de données, excepté dans les bases simulées 1, 2 et 4, Iris setosa, Biomed Healthy, Diabetes Present, spectf 1 et Glass Building Fload, où l'on observe une diminution de cet indice.

La F-mesure diminue sensiblement dans les bases Ecoli Periplasm et Balance-Scale Lef (respectivement 0.576 et 0.592) pour l'approche ACP. On observe également une très faible diminution des performances de l'approche One-SVM dans les bases HeptaM, GolfballM et Balance-Scale Left.

Indice de AUC : pour l'indice AUC représenté dans le tableau 3.8, GOF-Novelty, ACP et One-SVM fournissent des résultats équivalents dans la majorité des bases de données, excepté pour la base Diabets Present, où l'on remarque une diminution importante de la valeur de l'AUC pour les approches ACP et One-SVM (0.431 et 0.52 respectivement) ; GOF-Novelty fournit pour cette même base une performance de 0.83.

Écart de la validation croisée : concernant l'analyse des écarts obtenus, les trois méthodes donnent des valeurs similaires. Dans la plupart des bases de données et des indices de performance, les valeurs obtenues restent largement équivalentes.

Cependant, certaines exceptions sont observées. En effet, la base de données anneauxM fournit un écart de l'indice du rappel de 0.1% pour notre approche GOF-Novelty. APC et One-SVM donnent respectivement 1.8 % et 2.1 %.

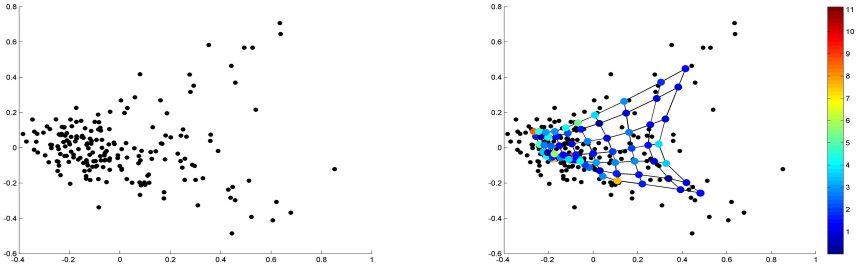
La même observation peut être faite pour l'écart de l'indice de précision de la base de données GolfballM, où GOF-Novelty fournit 0.9%, les approches ACP et One-SVM fournissent 9.3 % et 2.1 % respectivement.

Bilan : dans la majorité des cas, notre méthode obtient de meilleurs résultats, surtout pour la F-mesure. Nous remarquons que notre algorithme est performant lorsque les bases de données contiennent des clusters denses avec beaucoup de données. C'est le cas de la base Golfball, où notre approche obtient les meilleurs performances sur les 4 indices. Par contre, dans la base simulée 1 ou 2, on a des nuages de point relativement éparpillés. C'est ce qui explique la baisse des performances obtenues.

Enfin, nous concluons, au regard de ces analyses réalisées sur différents indices de performance, que GOF-Novelty est une approche qui permet de détecter les nouveautés d'une manière pertinente. Notre approche donne dans la plupart des bases de données des résultats stables par rapport aux approches ACP et One-SVM.

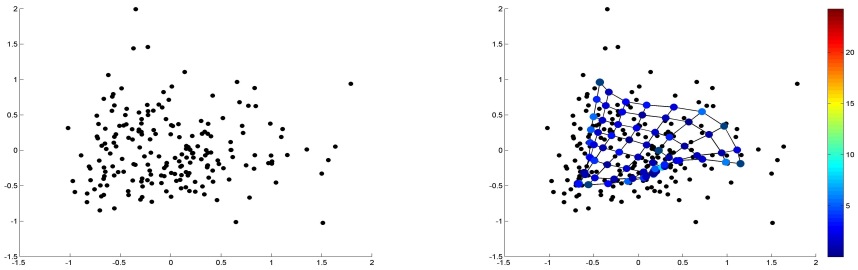
3.4.2 Résultats visuels des bases réelles

Nous avons projeté les données dans un plan à 2 dimensions afin de visualiser les données ainsi que la carte GOF-SOM. Les projections des données (sous-figures de gauche) montrent qu'aucune des bases réelles ne possède des groupes isolés, excepté la base glass. Nous remarquons que le score d'outlierness des référents, représenté par des couleurs (bleu = cluster normal, rouge = cluster outlier) reste relativement homogène dans la plupart des bases de données. Lorsque quelques groupes de données sont denses et relativement isolés, le score GOF est relativement élevé. C'est le cas des bases de données glass (groupe de données en bas à gauche) et Iris S (groupe de données à gauche).



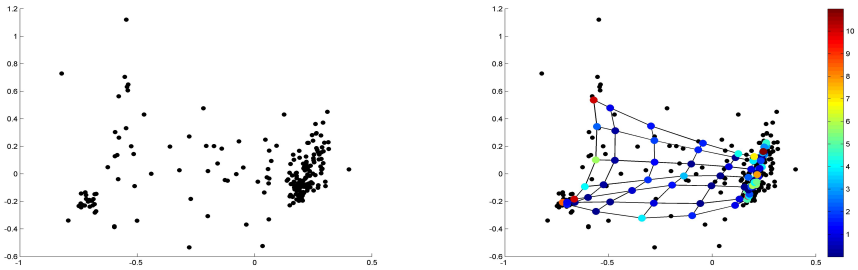
(a) Données de la base BiomedHealthy (b) Carte associée aux données de la base BiomedHealthy

FIGURE 3.12 – GOF-SOM appliqué sur les données de la base BiomedHealthy



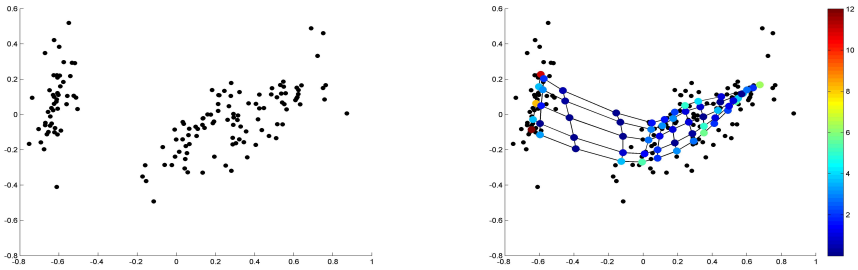
(a) Données de la base CancerWpbcRet (b) Carte associée aux données de la base CancerWpbcRet

FIGURE 3.13 – GOF-SOM appliqué sur les données de la base CancerWpbcRet



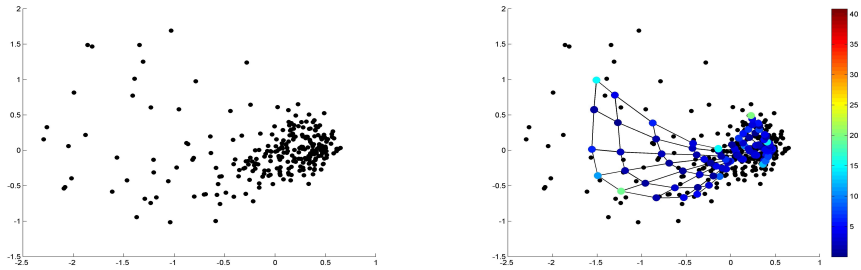
(a) Données de la base GlassBuilding-Float (b) Carte associée aux données de la base GlassBuildingFloat

FIGURE 3.14 – GOF-SOM appliqué sur les données de la base GlassBuilding-Float



(a) Données de la base IrisVirginica (b) Carte associée aux données de la base IrisVirginica

FIGURE 3.15 – GOF-SOM appliqué sur les données de la base IrisVirginica



(a) Données de la base Spectf1

(b) Carte associée aux données de la base Spectf1

FIGURE 3.16 – GOF-SOM appliqué sur les données de la base Spectf1

3.5 Conclusion

Nous nous sommes intéressé dans la première partie de ce travail au problème de détection de groupes-outliers. Nous avons présenté un nouveau score (paramètre) GOF qui est basé sur les densités locales des clusters. Ce score a été intégré aux cartes auto-organisatrices. Une série d'expériences a été réalisée pour valider la méthode proposée. Les résultats obtenus ont été analysés visuellement et analytiquement. Ceci nous a permis de mieux évaluer notre approche, qui s'est avérée prometteuse comme solution au problème de détection de groupes-outliers.

Dans la seconde partie, nous avons utilisé GOF comme classifieur pour le problème de la détection des nouveautés. À notre connaissance, détecter des groupes-outliers dans un cadre d'apprentissage non supervisé en mesurant un score "d'outlier-ness", ensuite, identifier et prédire les nouveautés en utilisant la détection des nouveautés est une tâche importante pour de nombreuses applications. Une série d'expérimentations a été réalisée afin de valider l'approche proposée. Nous avons comparé les performances de notre approche avec deux méthodes classiques de détection des nouveautés. Les résultats expérimentaux montrent que notre approche est prometteuse et qu'elle permet d'identifier les données nouvelles.

Dans la continuité de nos travaux sur la détection de groupes "outliers", nous souhaitons définir un nouveau score qui prenne en compte à la fois les observations et les variables dans le processus d'apprentissage. C'est ainsi que nous nous sommes intéressé à la classification croisée (ou bi-partitionnement). Comment peut-on concevoir un système qui permet de générer un partitionnement simultané des lignes (observations) et des colonnes (variables), tout en proposant un score mesurant l'intérêt d'un bloc d'observations/variables ?

À cet effet, nous proposons un nouveau modèle, qui utilise les cartes auto-organisatrices pour organiser la matrice des données en blocs (bi-clusters) homogènes, tout en prenant en compte simultanément les lignes et les colonnes. Ce modèle fera l'objet de la seconde contribution de cette thèse, qui est détaillée dans le chapitre suivant.

Chapitre 4

Contribution : bi-partitionnement topologique

Sommaire

Dans cette partie de travail, nous proposons une nouvelle approche (BiTM) de bi-partitionnement utilisant les cartes topologiques. BiTM ne nécessite aucune pré-organisation de la matrice des données. Notre modèle utilise une seule carte, qui représente simultanément la partition des observations et la partition des variables. Notre approche permet aussi de fournir de nouvelles visualisations.

4.1 Modèle proposé : approche de bi-partitionnement utilisant les cartes topologiques (BiTM)

Le modèle BiTM (Bi-clustering using Topological Maps) est constitué d'un ensemble de cellules discrètes $\mathcal{C}$ de taille K appelées "cartes". Ces cartes ont une topologie discrète définie comme un graphe non orienté, qui est généralement une grille à 2 dimensions. Pour chaque paire de cellules (c, r) de la carte, la distance $\delta(c, r)$ est définie par le plus court chemin reliant les cellules r et c sur la grille. Soit $\Re^d$ l'espace euclidien des données et $\mathcal{A}$ la matrice des données, où chaque observation $\mathbf{x}_i = (x_i^1, x_i^2, \ldots, x_i^j, \ldots, x_i^d)$ est un vecteur dans $\Re^d$.

L'objectif de BiTM est de fournir des bi-clusters organisés dans une carte topologique. Afin de montrer le lien avec l'approche Croeuc [Govaert 1983], nous avons choisi d'utiliser les mêmes notations que celles introduites par les travaux de [Govaert 1983]. Pour cela, l'ensemble des lignes (observations) $I = \{1, \ldots, N\}$ de la matrice des données $\mathcal{A}$ est partitionné en K groupes $\{P_1, P_2, \ldots, P_k, \ldots, P_K\}$ où $P_k = \{\mathbf{x}_i, \phi_z(\mathbf{x}_i) = k\}$ et $\phi_z(.)$ représente la fonction d'affectation spécifique aux lignes. De même, l'ensemble des colonnes (variables) $J = \{1, \ldots, d\}$ est partitionné en L groupes $\{Q_1, Q_2, \ldots, Q_l, \ldots, Q_L\}$ où $Q_l = \{\mathbf{x}^j, \phi_z(\mathbf{x}^j) = l\}$ et $\phi_w(.)$ représente la fonction d'affectation spécifique aux colonnes.

Nous définissons par la suite deux matrices binaires $Z = (z_{ik})$ et $W = (w_{jl})$ pour sauvegarder les informations, associées respectivement aux observations et aux variables.

$$z_{ik} = \begin{cases} 1 & \text{si } \mathbf{x}_i \in P_k, k = \phi_z(\mathbf{x}_i) \\ 0 & \text{sinon} \end{cases}$$

$$w_{jl} = \begin{cases} 1 & \text{si } \mathbf{x}^j \in Q_l, l = \phi_w(\mathbf{x}^j) \\ 0 & \text{sinon} \end{cases}$$

Avec z_{ik} et w_{jl}, nous pouvons déterminer des blocs de données $B_k^l = \{x_{ij} | z_{ik} \times w_{jl} = 1\}$. Dans BiTM, chaque cellule k de $\mathcal{C}$ est associée à un prototype sous la forme d'un vecteur : $\mathbf{g}_k = (g_k^1, g_k^2, \ldots, g_k^l, \ldots, g_k^L)$ de dimension $L < d$ où g_k^l est le prototype du bloc B_k^l. Dans le cadre des cartes topologiques, nous proposons de minimiser la nouvelle fonction de coût suivante :

$$\mathcal{J}_{BiTM}(\phi_w, \phi_z, G) = \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{d} \sum_{r=1}^{K} \mathcal{K}^T(\delta(r, k)) w_{jl} \times z_{ik} (x_i^j - g_r^l)^2 \quad (4.1)$$

Elle peut être réécrite de la manière suivante :

$$\mathcal{J}_{BiTM}(\phi_w, \phi_z, G) = \sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{\mathbf{x}_i \in P_k}\sum_{\mathbf{x}^j \in Q_l}\sum_{r=1}^{K} \mathcal{K}^T(\delta(r,k))\ (x_i^j - g_r^l)^2 \qquad (4.2)$$

ϕ_z la fonction d'affectation des lignes (observations).
ϕ_w la fonction d'affectation des colonnes (variables).
$G = \{\mathbf{g}_1, \ldots, \mathbf{g}_K\}$ désigne l'ensemble des prototypes.
$\mathcal{K}^T(\delta(r,k))$ la fonction de voisinage.
T permet de définir le rayon de voisinage.
De même que pour les cartes auto-organisatrices, nous utilisons la fonction $\mathcal{K}^T(\delta(c,r)) = \exp(\frac{-\delta(c,r)}{T})$ pour définir le voisinage.

L'expression 4.2 représente une famille de fonction de coût paramétrée par T. Elle est une extension de la fonction Croeuc [1] dans laquelle la distance est remplacée par une distance pondérée par la fonction de voisinage $\mathcal{K}^T(\delta(r,k))$.

La méthode utilisée pour la minimisation de la fonction de coût $\mathcal{J}_{BiTM}$ à T fixe utilise le formalisme des nuées dynamiques qui peut assurer une convergence vers un minimum local de la fonction $\mathcal{J}_{BiTM}$. La minimisation de la fonction $\mathcal{J}_{BiTM}$, pour une valeur de T fixe, est donc réalisée comme dans le cas de l'algorithme Croeuc, par itérations successives, chacune se décompose de 3 phases :

1. **Phase d'affectation des observations :** cette phase minimise la fonction $\mathcal{J}_{BiTM}$ par rapport à la fonction d'affectation des observations $\phi_z(\mathbf{x_i})$ en supposant la fonction d'affectation des variables $\phi_w(\mathbf{x^j})$ et l'ensemble des référents G fixés à la valeur courante. L'affectation qui minimise $\mathcal{J}_{BiTM}$ pour $\phi_w(\mathbf{x^j})$ et G fixés est définie pour chaque observation $\mathbf{x_i}$ par :

$$\phi_z(\mathbf{x}_i) = \arg\min_c \sum_{j=1}^{d}\sum_{l=1}^{L}\sum_{r=1}^{K} w_{jl} \times \mathcal{K}^T(\delta(r,c)) \times (x_i^j - g_r^l)^2$$

   Cette phase permet de définir une fonction d'affectation et une partition des lignes des données de $\mathcal{A}$.

2. **Phase d'affectation des variables :** cette phase minimise la fonction $\mathcal{J}_{BiTM}$ par rapport à la fonction d'affectation des variables $\phi_w(\mathbf{x^j})$ en supposant la fonction d'affectation des observations $\phi_z(\mathbf{x_i})$ et l'ensemble des référents G fixés à la valeur courante. L'affectation qui minimise $\mathcal{J}_{BiTM}$ pour $\phi_z(\mathbf{x_i})$ et G fixés est définie pour chaque variable $\mathbf{x^j}$ par :

$$\phi_w(\mathbf{x}^j) = \arg\min_l \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{r=1}^{K} z_{ik} \times \mathcal{K}^T(\delta(r,k)) \times (x_i^j - g_r^l)^2$$

1. Voir la section 1.4.1 pour les détails de l'algorithme Croeuc

Cette phase permet de définir une fonction d'affectation et une partition des colonnes des données de $\mathcal{A}$.

3. **Phase de minimisation :** il s'agit maintenant de minimiser la fonction $\mathcal{J}_{BiTM}$ par rapport à l'ensemble des référents G, en supposant que $\phi_z(\mathbf{x_i})$ et $\phi_w(\mathbf{x^j})$ sont fixées à la valeur courante. $\phi_z(\mathbf{x_i})$ et $\phi_w(\mathbf{x^j})$ fixées, la fonction $\mathcal{J}_{BiTM}$ est quadratique par rapport à $\mathcal{W}$, elle admet donc un minimum unique qui est atteint pour $\frac{\partial \mathcal{J}_{BiTM}}{\partial g_r^l} = 0$. Les nouveaux vecteurs référents sont alors définis par la formule suivante :

$$g_r^l = \frac{\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{d} \mathcal{K}^T(\delta(r,k))) \times w_{jl} \times z_{ik} \times x_i^j}{\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{d} \mathcal{K}^T(\delta(r,k))) \times w_{jl} \times z_{ik}}$$

Chaque observation $\mathbf{x}_i$ appartient à une seule cellule k de la carte C. Ainsi, il est possible de réécrire l'expression ci-dessus de la manière suivante :

$$g_r^l = \frac{\sum_{i=1}^{N}\sum_{j=1}^{d} \mathcal{K}^T(\delta(r,\phi_z(\mathbf{x}_i))) \times w_{jl} \times x_i^j}{\sum_{i=1}^{N}\sum_{j=1}^{d} \mathcal{K}^T(\delta(r,\phi_z(\mathbf{x}_i))) \times w_{jl}}$$

La minimisation de $\mathcal{J}_{BiTM}$ s'effectue par itération successive jusqu'à stabilisation des 3 phases ou jusqu'à atteindre un nombre d'itérations N_{iter} défini à l'avance.
La formule la plus souvent utilisée pour faire décroître T est la suivante : $T = T_{max}\left(\frac{T_{min}}{T_{max}}\right)^{\frac{t}{N_{iter}-1}}$, où T^{max} représente la température initiale et T^{min} la température finale atteinte à l'itération N_{iter}.

L'initialisation des référents G de la carte BiTM peut être réalisée par un algorithme de partitionnement simple sur les lignes et sur les colonnes de la matrice de données (dans notre cas, un K-means sur les lignes et sur les colonnes).

L'algorithme global pour une fonction particulière de décroissance T est présenté dans le paragraphe suivant.

Algorithme 12

Entrées :

- Les données $\mathcal{A} = \{x_i^j\}_{i=1...N, j=1...d}$.
- Les matrices d'affectation Z, W.
- Les prototypes G de la carte initialisés.
- N_{iter} : le nombre maximum d'itérations.

Sorties :

- Les matrices d'affectation Z, W.
- Les prototypes G mis à jour.

Phase itérative

1- Affectation des observations : chaque observation $\mathbf{x}_i$ est affectée au prototype $\mathbf{g}_k$ le plus proche en utilisant la fonction d'affectation :

$$\phi_z(\mathbf{x}_i) = \arg\min_c \sum_{j=1}^{d} \sum_{l=1}^{L} \sum_{r=1}^{K} w_{jl} \times \mathcal{K}^T(\delta(r,c)) \times (x_i^j - g_r^l)^2$$

2- Mise à jour des prototypes : les vecteurs des prototypes sont mis à jour en fonction des affectations des observations :

$$g_r^l = \frac{\sum_{i=1}^{N} \sum_{j=1}^{d} \mathcal{K}^T(\delta(k, \phi_z(\mathbf{x}_i)) \times w_{jl} \times x_i^j}{\sum_{i=1}^{N} \sum_{j=1}^{d} \mathcal{K}^T(\delta(k, \phi_z(\mathbf{x}_i))) \times w_{jl}}$$

3- Affectation des variables : chaque variable $\mathbf{x}^j$ est affectée au prototype $\mathbf{g}_k^l$ le plus proche en utilisant la fonction d'affectation :

$$\phi_w(\mathbf{x}^j) = \arg\min_l \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{r=1}^{K} z_{ik} \times \mathcal{K}^T(\delta(r,k)) \times (x_i^j - g_r^l)^2$$

4- Mise à jour des prototypes : les vecteurs des prototypes sont mis à jour en fonction des affectations des variables :

$$g_r^l = \frac{\sum_{i=1}^{N} \sum_{j=1}^{d} \mathcal{K}^T(\delta(k, \phi_z(\mathbf{x}_i))) \times w_{jl} \times x_i^j}{\sum_{i=1}^{N} \sum_{j=1}^{d} \mathcal{K}^T(\delta(k, \phi_z(\mathbf{x}_i))) \times w_{jl}}$$

Répéter les phases 1, 2, 3 et 4 jusqu'à N_{iter} qui représente le nombre d'itérations.

4.1.1 L'ordre topologique dans le modèle BiTM

Comme au paragraphe 1.3.4.3, la décomposition de la fonction de coût $\mathcal{J}_{BiTM}$, qui dépend de la valeur de T, peut être réécrite de la manière suivante :

$$\begin{aligned}\mathcal{J}_{BiTM}(\phi_w,\phi_z,G) &= \sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{\mathbf{x}_i\in P_k}\sum_{\mathbf{x}^j\in Q_l}\sum_{r=1,r\neq k}^{K}\mathcal{K}^T(\delta(r,k))(x_i^j-g_r^l)^2 \\ &+ \sum_{r=1}^{K}\sum_{l=1}^{L}\sum_{\mathbf{x}_i\in P_k}\sum_{\mathbf{x}^j\in Q_l}K^T(\delta(r,r))(x_i^j-g_r^l)^2\end{aligned}$$

$$\begin{aligned}\mathcal{J}_{BiTM}(\phi_w,\phi_z,G) &= \sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{\mathbf{x}_i\in P_k}\sum_{\mathbf{x}^j\in Q_l}\sum_{r=1,r\neq k}^{K}\mathcal{K}^T(\delta(r,k))(x_i^j-g_r^l)^2 \\ &+ \mathcal{K}^T(0)\sum_{r=1}^{K}\sum_{l=1}^{L}\sum_{\mathbf{x}_i\in P_k}\sum_{\mathbf{x}^j\in Q_l}(x_i^j-g_r^l)^2\end{aligned}$$

Nous remarquons que le terme : $\sum_{r=1}^{K}\sum_{l=1}^{L}\sum_{\mathbf{x}_i\in P_k}\sum_{\mathbf{x}^j\in Q_l}(x_i^j-g_r^l)^2$ représente la fonction de coût de l'algorithme Croeuc, sachant que $K^T(\delta(r,r)) = K^T(0)$

$$\mathcal{J}_{Crouec}(\phi_w,\phi_z,G) = \sum_{r=1}^{K}\sum_{l=1}^{L}\sum_{\mathbf{x}_i\in P_k}\sum_{\mathbf{x}^j\in Q_l}(x_i^j-g_r^l)^2$$

L'expression $\mathcal{J}_{BiTM}$ définie sous cette forme permet de mener une discussion similaire à celle proposée au paragraphe 1.3.4.3 du chapitre 1. L'utilisation de grande valeur de T permet une action sur le premier terme qui introduit l'ordre topologique. Quand T tend vers zéro c'est le second terme qui agit et minimise l'inertie locale. La fonction de coût $\mathcal{J}_{BiTM}$ est décomposée en deux termes. Afin de maintenir l'ordre topologique entre les blocs, la minimisation du premier terme entraîne le bloc qui correspond à deux cellules voisines. En effet, si les cellules c et r sont voisines dans la carte, la valeur de $\delta(r,k)$ est faible, et dans ce cas, la valeur de $\mathcal{K}^T(\delta(r,k))$ est élevée. La minimisation du second terme correspond à la minimisation de l'inertie des données locales affectées à un bloc $B_r^j, j = 1\ldots L$. Pour différentes valeurs de T, chaque terme de la fonction de coût a une importance relative dans le processus de minimisation. On peut, donc, définir deux étapes pour l'exploitation de l'algorithme :

- La première étape correspond à des valeurs élevées de T. Si le premier terme est dominant, alors la priorité est de préserver la topologie.
- La deuxième étape correspond à des valeurs faibles de T, où le deuxième terme est pris en compte dans la fonction de coût. Par conséquent,

l'adaptation locale et l'algorithme BiTM convergent vers l'algorithme Crouec [2] proposé par [Govaert 1983].

Nous considérons alors que la première phase est une étape d'initialisation de la deuxième phase, ce qui permet d'assurer une conservation de la topologie et d'obtenir une partition sur les lignes et les colonnes en sous-ensembles "homogènes".

4.2 Expérimentations

4.2.1 Mesures de performances

Afin d'évaluer la qualité de notre approche, nous avons sélectionné 3 mesures de performances externes : pureté, rand et NMI ([Strehl 2002]). Pour calculer ces critères, nous considérons un ensemble de N objects avec L classes classifiées en K clusters et deux partitions pour comparer : $X = \{\mathbf{x}_1, .., \mathbf{x}_N\}$ où $\mathbf{x}_k \in [C_1..C_K]$ est une variable aléatoire d'affectation et $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ où $\mathbf{y}_l = [B_1..B_L]$ est une variable aléatoire pour les étiquettes pré-existantes. Par conséquent, le tableau de contingence (tableau 4.1) peut être exprimé comme suit :

R\C	C_1	C_2	$\cdots$	C_k	$\cdots$	C_K	Somme
B_1	n_{11}	n_{12}	$\cdots$	n_{1k}	$\cdots$	n_{1K}	N_{1*}
B_2	n_{21}	n_{22}	$\cdots$	n_{2k}	$\cdots$	n_{2K}	N_{2*}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
B_l	n_{l1}	n_{l2}	$\cdots$	n_{lk}	$\cdots$	n_{lK}	N_{l*}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
B_L	n_{L1}	n_{L2}	$\cdots$	n_{Lk}	$\cdots$	n_{LK}	N_{L*}
Somme	N_{*1}	N_{*2}		N_{*k}		N_{*K}	N

TABLE 4.1 – Tableau de contingence

L'indice de pureté est calculé comme suit :

$$I = \frac{\sum_k max_{l=[1..L]}(n_{lk})}{N} \tag{4.3}$$

L'indice de rand est donné par la formule suivante :

$$Rand = (N_{00} + N_{11}) / \left(\tfrac{N}{2} \right) \tag{4.4}$$

Où

2. Voir la section 1.4.1 pour les détails de l'algorithme Croeuc

– N_{11} est le nombre de paires qui sont dans le même groupe que B et C ;
– N_{00} est le nombre de paires qui sont différentes dans B et C.

L'indice de NMI (Normalized Mutual Information) est donné par l'expression suivante :

$$NMI = \frac{\sum_{l}\sum_{k} n_{l_k} log_2(\frac{N.n_{l,k}}{N_{l*}N_{*k}})}{(\sum_{l} N_{l*} log_2(\frac{N_{l*}}{N})(\sum_{k} N_{*k} log_2(\frac{N_{*k}}{N})))} \tag{4.5}$$

4.2.2 Description des bases de données utilisées

Nous avons testé l'algorithme BiTM avec des jeux de données du répertoire UCI ([Frank 2010]). Nous avons aussi utilisé dans ces expérimentations des bases de données binaires synthétiques [3]. La particularité de ces bases de données est que les observations et les variables sont étiquetées. Nous allons particulièrement utiliser ces bases de données afin d'évaluer le clustering des variables. Les tableaux 4.2 et 4.3 indiquent les paramètres de chaque jeu de données (nombre d'observations, nombre de variables, taille de la carte utilisée pour l'apprentissage et nombre de classes réelles).

Bases de données	# Observations	# Variables	Taille carte	# Classes
isolet5	1559	617	12×12	26
Movement Libras	45	90	5× 5	15
Breast	699	10	7×7	2
Sonar Mines	208	60	6×6	2
Lung Cancer	32	56	4×4	2
Spectf 1	349	44	4×4	2
Cancer Wpbc Ret	198	33	6×6	2
Horse Colic	300	27	5×5	2
Heart	270	13	5×5	2
glass	214	9	5×5	7

TABLE 4.2 – Description des jeux de données d'UCI.

3. Ces bases de données sont fournies par Lazhar Labiod https ://sites.google.com/site/lazharlabiod/

Bases de données	# Observations	# Variables	Taille carte	# Classes
Simulé 1	2000	5000	12×12	3
Simulé 2	2000	5000	10×10	3
Simulé 3	2000	5000	8×8	3
Simulé 4	2000	5000	6×6	3

TABLE 4.3 – Description des jeux de données simulés.

4.2.3 Comparaison de BiTM avec les approches de partitionnement

Protocole de validation

Dans cette première expérimentation, nous comparons les résultats de notre approche BiTM avec les approches suivantes : SOM classique (Self Organizing Maps [Kohonen 2001]), HCL (Hierarchical Clustering [Eisen 1998]) et NMF (Non-Negative Matrix Factorization [Paatero 1994]). Notre objectif à travers cette comparaison est de montrer que BiTM ne modifie pas le fonctionnement général des cartes auto-organisatrices et fournit des performances comparables aux algorithmes classiques de partitionnement. Nous présentons dans les tableaux 4.4, 4.5 et 4.6 les résultats obtenus en termes d'indices de pureté, rand et NMI.

Nous avons choisi la taille de les cartes BiTM et SOM selon l'heuristique de Kohonen ($5 \times K^{0.54321}$). Nous avons choisi le nombre de clusters des observations proportionnellement au nombre de cellules non vides dans BiTM et SOM. L'initialisation des partitions des lignes est effectuée d'une manière aléatoire pour l'ensemble des approches BiTM, SOM, HCL et NMF. Nous avons normalisé l'ensemble des jeux de données entre 0 et 1. Nous avons calculé 3 indices de performances (pureté, rand et NMI) sur l'ensemble des résultats obtenus avec les approches BiTM, SOM, HCL et NMF. Nous avons sélectionné la meilleure performance obtenue dans les 10 expérimentations réalisées.

Bases de données	BiTM	SOM	HCL	NMF
isolet5	0.316	**0.433**	0.427	0.107
Movement Libras	**0.712**	0.711	0.711	0.288
Breast	**0.978**	0.974	0.633	0.804
Sonar Mines	**0.769**	0.744	0.727	0.601
Lung Cancer	**1**	0.906	0.743	0.781
Spectf 1	**0.759**	0.716	0.65	0.73
Cancer Wpbc Ret	0.787	**0.828**	0.722	0.762
HorseColic	0.719	**0.78**	0.713	0.67
Heart	**0.883**	0.851	0.755	0.62
glass	0.618	0.623	**0.72**	0.481

TABLE 4.4 – Partitionnement : résultats de l'indice de pureté obtenus avec BiTM, SOM, HCL et NMF

Bases de données	BiTM	SOM	HCL	NMF
isolet5	**0.926**	0.905	0.812	0.471
Movement Libras	0.937	**0.943**	0.817	0.789
Breast	**0.687**	0.476	0.499	0.545
Sonar Mines	**0.508**	0.507	0.489	0.504
Lung Cancer	0.459	0.425	0.427	**0.487**
Spectf 1	0.418	0.403	**0.499**	0.436
Cancer Wpbc Ret	0.435	0.372	**0.54**	0.417
HorseColic	0.472	0.448	0.449	**0.462**
Heart	**0.56**	0.529	0.512	0.502
glass	0.653	**0.752**	0.348	0.689

TABLE 4.5 – Partitionnement : résultats de l'indice de rand obtenus avec BiTM , SOM, HCL et NMF

Bases de données	BiTM	SOM	HCL	NMF
isolet5	0.439	**0.584**	0.562	0.007
Movement Libras	**0.811**	0.797	0.555	0.57
Breast	**0.53**	0.364	0.003	0.193
Sonar Mines	0.158	**0.233**	0.001	0.026
Lung Cancer	**0.461**	0.344	0.295	0.111
Spectf 1	0.1449	**0.185**	0.01	0.025
Cancer Wpbc Ret	0.081	**0.14**	0.005	0.014
HorseColic	0.06	**0.128**	0.009	0.03
Heart	**0.247**	0.225	0.06	0.04
glass	0.125	**0.463**	0.153	0.231

TABLE 4.6 – Partitionnement : résultats de l'indice de NMI obtenus avec BiTM , SOM, HCL et NMF

Indice de pureté : le tableau 4.4 présente les résultats expérimentaux obtenus avec les approches BiTM, SOM, HCL et NMF sur l'indice de pureté. Notre approche BiTM donne des résultats meilleurs ou équivalents dans la plupart des bases de données. Nous remarquons une légère baisse de performance sur les bases : Horse Colic, Cancer Wpbc Ret, glass et isolet5.

Indice de rand : en ce qui concerne le tableau 4.5, BiTM, comparé à SOM, HCL et NMF, donne des résultats meilleurs ou équivalents en termes d'indice de rand dans les bases isolet5, Movement Libras, Breast, Sonar Mines, Horse colic et heart. Même si notre méthode est moins efficace avec les bases Lung cancer, Spectf 1, Cancer Wpbc Ret et glass, BiTM reste stable. Par exemple, avec la base Movement Libras, la valeur la plus élevée de l'indice de rand est donnée par SOM 0.943. Notre méthode obtient 0.937. Cependant, HCL obtient une performance de 0.817 et NMF 0.789. La même analyse peut être faite sur les autres bases de données.

Indice de NMI : l'analyse des résultats de l'indice NMI présenté dans le tableau 4.6 montre que BiTM, SOM et HCL fournissent des performances équivalentes en termes d'indice NMI. Notre méthode est meilleure ou équivalente à l'approche SOM dans les bases Movement Libras, Breast, Lung cancer, Spectf 1 et Heart. HCL obtient une très forte diminution de l'indice NMI dans la plupart des bases, à l'exception de isolet5, Movement Libras, Lung cancer et glass. En dépit d'une faible diminution de l'indice NMI dans certaines bases de données, notre approche fournit des résultats stables pour l'ensemble des bases de données.

4.2.4 Comparaison de BiTM avec les approches de bi-partitionnement

Protocole de validation

Afin de comparer BiTM avec les approches de bi-partitionnement, nous avons sélectionné les approches suivantes : CTWC ([Getz 2000a]), NBVD ([Long 2005]) et CUNMTF ([Labiod 2011]). Les résultats expérimentaux sont présentés dans les tableaux 4.7, 4.8 et 4.9. Nous avons choisi la taille de la carte BiTM selon l'heuristique de Kohonen ($5 \times K^{0.54321}$). Le nombre de clusters des variables (colonnes de la matrice $\mathcal{A}$) est exactement le même pour l'ensemble des approches BiTM, CTWC, NBVD et CUNMTF. Cependant, pour le nombre de clusters des observations (lignes de la matrice $\mathcal{A}$), nous avons pris une taille proportionnelle au nombre de cellules non vides dans BiTM. Par exemple, dans le cas de la base Lung Cancer, la taille de la carte la taille de la carte BiTM est égale $4 \times 4 = 16$. Le nombre de cellules vides de la carte BiTM est égale à 4. Ainsi, la taille de la partition des lignes que nous

avons choisi pour les approches CTWC, NBVD et CUNMTF est égale à 16-4 = 12. L'initialisation des partitions des lignes et des colonnes est effectuée d'une manière aléatoire pour l'ensemble des approches BiTM, CTWC, NBVD et CUNMTF. Nous avons normalisé l'ensemble des jeux de données entre 0 et 1. Nous avons calculé 3 indices de performances (pureté, rand et NMI) sur l'ensemble des résultats obtenus avec les approches BiTM, CTWC, NBVD et CUNMTF. Nous avons sélectionné la meilleure performance obtenue dans les 10 expérimentations réalisées.

Bases de données	BiTM	CTWC	NBVD	CUNMTF
isolet5	**0.316**	0.103	0.073	0.293
Movement Libras	**0.712**	NaN	0.33	0.333
Breast	**0.978**	0.655	0.834	0.834
Sonar Mines	**0.769**	0.548	0.644	0.634
Lung Cancer	**1**	0.718	0.875	0.843
Spectf 1	**0.759**	0.727	0.727	0.727
Cancer Wpbc Ret	**0.787**	0.762	0.762	0.762
HorseColic	**0.719**	0.67	0.67	0.673
Heart	**0.883**	0.555	0.674	0.674
glass	**0.618**	0.523	0.462	0.462

TABLE 4.7 – Bi-partitionnement : comparaison en utilisant l'indice de pureté obtenu avec BiTM, CTWC, NBVD et CUNMTF.

Bases de données	BiTM	CTWC	NBVD	CUNMTF
isolet5	**0.926**	0.91	0.502	0.508
Movement Libras	**0.937**	NaN	0.845	0.84
Breast	0.687	0.505	0.659	**0.688**
Sonar Mines	**0.508**	0.502	0.514	**0.508**
Lung Cancer	0.459	**0.556**	**0.556**	0.536
Spectf 1	0.418	**0.513**	0.42	0.42
Cancer Wpbc Ret	0.435	**0.524**	0.414	0.417
HorseColic	**0.472**	0.463	0.46	0.459
Heart	**0.56**	0.498	0.513	0.515
glass	0.653	0.69	**0.693**	0.69

TABLE 4.8 – Bi-partitionnement : comparaison en utilisant l'indice de rand obtenu avec BiTM, CTWC, NBVD et CUNMTF.

Bases de données	BiTM	CTWC	NBVD	CUNMTF
isolet5	**0.439**	0.077	0.137	0.186
Movement Libras	**0.811**	NaN	0.688	0.667
Breast	**0.53**	0.01	0.243	0.0233
Sonar Mines	**0.158**	0.006	0.057	0.04
Lung Cancer	**0.461**	0.041	0.309	0.261
Spectf 1	**0.1449**	0.001	0.016	0.016
Cancer Wpbc Ret	**0.081**	0.034	0.024	0.031
HorseColic	**0.06**	0.003	0.03	0.028
Heart	**0.247**	0.001	0.063	0.061
glass	0.125	**0.38**	0.246	0.245

TABLE 4.9 – Bi-partitionnement : comparaison en utilisant l'indice NMI obtenu avec BiTM, CTWC, NBVD et CUNMTF.

Indice de pureté : le tableau 4.7 résume les résultats expérimentaux de l'indice de pureté. Nous remarquons que BiTM fournit les meilleurs résultats sur toutes les bases de données. Dans la plupart des cas, nous constatons une différence remarquable entre les résultats sur l'indice de pureté obtenu avec notre méthode et les autres approches. En effet, pour la base Movement Libras, par exemple, BiTM obtient 0.712, NBVD 0.33 et CUNMTF 0.333. La même constatation est valable pour la base Lung Cancer, où BiTM obtient 1, CTWC 0.718, NBVD 0.875 et CUNMTF 0.843. Nous observons aussi la difficulté d'obtenir de grandes valeurs de l'indice de pureté pour la base isolet5.

Indice de rand : comme indiqué dans le tableau 4.8, BiTM fournit un indice de rand similaire et même meilleur que celui obtenu par les autres méthodes dans la majorité des cas.

Indice de NMI : le tableau 4.9 présente les résultats expérimentaux obtenus avec BiTM, CTWC, NBVD et CUNMTF pour l'indice NMI. Notre approche BiTM fournit les plus hautes valeurs de l'indice NMI pour toutes les bases de données, excepté la base glass.

4.2.5 Cas particulier : comparaison des performances de BiTM avec les approches de bi-partitionnement sur les bases de données simulées binaires

Dans les bases de données réelles, il est très difficile d'obtenir les étiquettes des classes des variables. Afin de valider le clustering des variables de notre modèle BiTM, nous avons utilisé des bases de données simulées étiquetées en ligne (observations) et en colonnes (variables) décrites dans le tableau 4.3. Les tableaux 4.10, 4.11 et 4.12 montrent les résultats obtenus des indices de pureté, rand et NMI pour le clustering des observations et des variables.

Nous constatons à travers les 3 indices de performance que notre approche est meilleure dans le cas du clustering des observations dans la plupart des bases de données. Cependant, nous remarquons une légère baisse des performances de BiTM au niveau du clustering des variables. Nous remarquons aussi que BiTM reste compétitif dans la majorité des cas, malgré cette légère baisse. Ces constatations sont justifiées du fait de l'utilisation des cartes topologiques comme algorithme d'apprentissage et du fait que les observations et les variables sont traitées de la même manière. Ceci engendre un déséquilibre des données.

Il est clair que nous allons améliorer le modèle BiTM afin d'obtenir de meilleures performances du clustering des variables. En premier lieu, nous allons adapter le modèle BiTM aux données binaires en utilisant une mesure de similarité adaptée. Dans le modèle BiTM, toutes les partitions sont constituées des mêmes blocs de variables. L'une des pistes de BiTM est justement la modification de cette contrainte de manière à retrouver, au niveau de chaque partition, des blocs de variables différents. Ceci représente un des axes des perspectives de nos travaux. Enfin, nous avons commencé à formaliser un nouveau modèle permettant de mesurer la pertinence de chaque bloc de variable obtenu par le modèle BiTM.

Bases	BiTM		CTWC		NBVD		CUNMTF	
	Observ	Var	Observ	Var	Observ	Var	Observ	Var
Simulé 1	**0.991**	**0.704**	0.901	0.512	0.828	0653	0.807	0.698
Simulé 2	0.881	**0.794**	**0.902**	0.612	0.712	0.783	0.515	0.804
Simulé 3	**0.999**	**0.793**	0.910	0.692	0.623	0.483	0.421	0.450
Simulé 4	0.881	**0.781**	**0.891**	0.732	0.391	0.802	0.488	0.694

TABLE 4.10 – Indice de pureté sur les bases simulées binaires des approches BiTM, CTWC, NBVD et CUNMTF.

Bases	BiTM		CTWC		NBVD		CUNMTF	
	Observ	Var	Observ	Var	Observ	Var	Observ	Var
Simulé 1	**0.930**	0.441	0.910	0.412	0.492	**0.510**	0.341	0.438
Simulé 2	**0.889**	0.370	0.718	**0.409**	0.831	0.352	0.582	0.375
Simulé 3	**0.875**	0.363	0.682	0.282	0.721	0.401	0.650	**0.625**
Simulé 4	**0.889**	0.370	0.812	**0.512**	0.691	0.290	0.620	0.440

TABLE 4.11 – Indice de rand sur les bases simulées binaires des approches BiTM, CTWC, NBVD et CUNMTF.

Bases	BiTM		CTWC		NBVD		CUNMTF	
	Observ	Var	Observ	Var	Observ	Var	Observ	Var
Simulé 1	**0.753**	0.020	0.312	**0.029**	0.421	0.021	0.312	0.010
Simulé 2	**0.754**	0.048	0.482	0.012	0.819	**0.123**	0.810	0.018
Simulé 3	**0.781**	0.057	0.584	**0.093**	0.321	0.022	0.608	0.015
Simulé 4	**0.754**	0.048	0.731	0.022	0.761	**0.099**	0.407	0.007

TABLE 4.12 – Indice de NMI sur les bases simulées binaires des approches BiTM, CTWC, NBVD et CUNMTF.

4.2.6 Apport pour l'analyse visuelle

Dans cette partie, nous montrons l'apport visuel de l'approche proposée. Notre approche BiTM se base sur les visualisations intuitives des cartes auto-organisatrices. Les figures 4.1, 4.2, 4.3, 4.4, 4.5 et 4.7 représentent différentes visualisations obtenues sur différentes bases de données (simulée1, isolet5, Movement Libras, Lung Cancer, Cancer Wpbc Ret, HorseColic et glass). Les figures 4.1(a), 4.2(a), 4.3(a), 4.4(a), 4.5(a) et 4.7(a) sont dédiées à la visualisation de la base de données organisé en fonction des groupes de lignes et de colonnes. Cette organisation est très claire dans le cas des bases binaires (voir la figure 4.1(a)).

Ces figures peuvent être obtenues par toute méthode de bi-partitionnement. Cependant, en utilisant cette visualisation, il est difficile d'analyser les blocs ou les bi-clusters obtenus. Afin de faciliter cette tâche, nous proposons de visualiser les bi-clusters en utilisant l'organisation topologique du modèle BiTM. Ainsi, chaque cellule de la carte est associée à la partition des observations et des variables. Cette organisation est illustrée par les figures 4.1(b), 4.2(c), 4.3(c), 4.4(c), 4.5(c) et 4.7(c) et par 4.2(d), 4.3(d), 4.4(d), 4.5(d) et 4.7(d) en organisant les cellules selon l'ordre des blocs de variables obtenus. Les figures 4.1(c) et 4.1(d) représentent les zoom des cellules 18 et 25 respectivement.

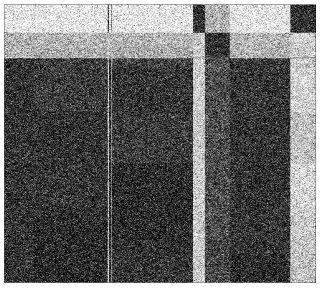
Dans le cas de la base Lung Cancer, par exemple, la figure 4.4(c) représente la carte topologique associée au modèle BiTM. Cette figure représente la topologie des groupes obtenus en appliquant l'algorithme BiTM. Nous remarquons une disposition des données au niveau de chaque cellule. Cette disposition est illustrée par une couleur. Plus la couleur est rouge, plus les variables ont de fortes valeurs.

Nous avons organisé la carte selon les blocs de variables obtenus. Le résultat est illustré dans la figure 4.4(d). Dans la première cellule, par exemple, nous remarquons que les variables ont changé de disposition de manière à créer une organisation au niveau de la cellule. Nous constatons clairement dans cette première cellule (en haut à gauche de la carte) que les blocs de

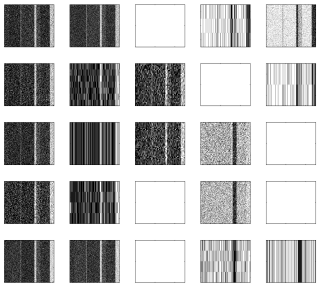
variables se comportent différemment à l'intérieur de cette cellule. Ce comportement est illustré par une couleur. Plus la couleur est rouge, plus le bloc de variables tend vers de fortes valeurs. Dans ce cas, nous remarquons que les premiers blocs de variables (totalement à gauche de la cellule) ont une couleur plutôt rouge. Par contre, le second bloc (au milieu de la cellule) est moins "important" car il est constitué de variables d'une couleur bleue. Enfin, le troisième bloc (totalement à droite) est constitué des variables moyennement importantes (couleur verte). Nous nous sommes focalisés dans cette analyse sur la base Lung Cancer. En fait, cette analyse peut être également réalisée sur les autres bases de données.

Visualisation de la distribution des blocs de variables : les figures 4.2(e), 4.3(e), 4.4(e), 4.5(e) et 4.7(e) représentent les blocs de variables sur la carte. Ces visualisations sont intéressantes, car elles permettent de représenter un résumé exhaustif de l'ensemble des variables. En effet, au lieu de visualiser toutes les variables des prototypes de la base Lung Cancer (56 variables), par exemple, on ne visualise que 4 blocs de variables. À partir de cette visualisation aussi, nous disposons d'une information sur la distribution des blocs de variables. Il est clair que les blocs 1 et 3 de la base Lung Cancer sont fortement corrélés.

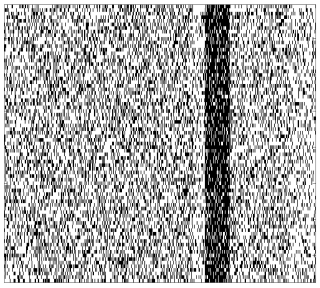
Finalement, BiTM a l'avantage de proposer une visualisation synthétique de la base de données et des bi-clusters. Ce résultat permet aux utilisateurs/experts une meilleure compréhension de la cohérence des données.



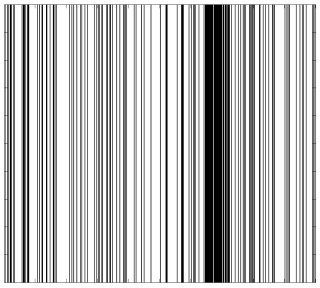
(a) La base de données organisée en fonction de l'ordre des observations et des variables de la classification croisée.



(b) Carte BiTM organisée selon les blocs de variables.

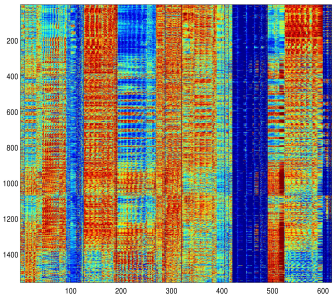


(c) Zoom sur la cellule numéro 18

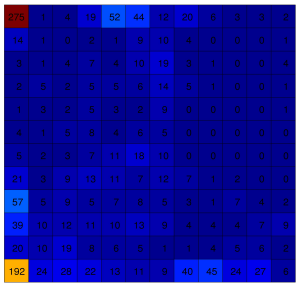


(d) Zoom sur la cellule numéro 25

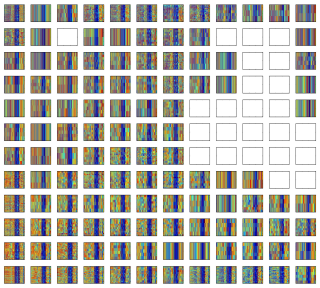
FIGURE 4.1 – Visualisation de la base de données binaire 1 en utilisant BiTM. Chaque cellule de la figure 4.1(b) indique une cellule de la carte.



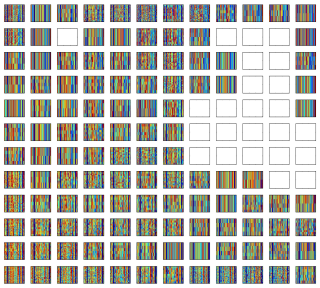
(a) La base de données organisée en fonction de l'ordre des observations et des variables de la classification croisée.



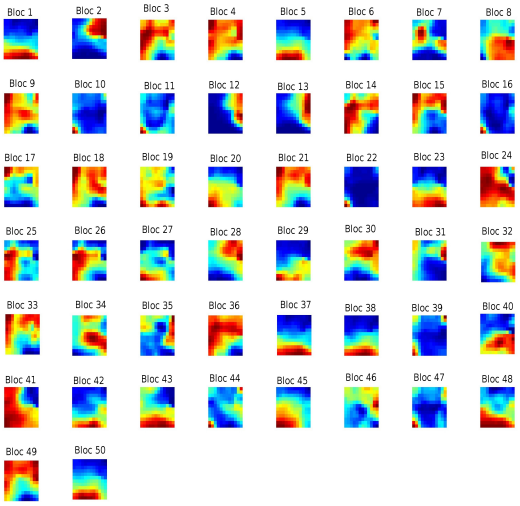
(b) Cardinalités de la carte.



(c) La carte BiTM. Représentation topologique des groupes de la carte BiTM.

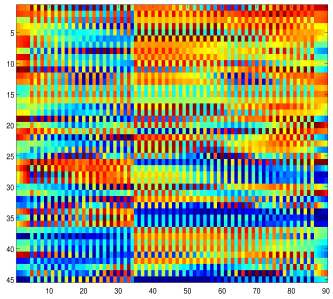


(d) Carte BiTM organisée selon les blocs de variables

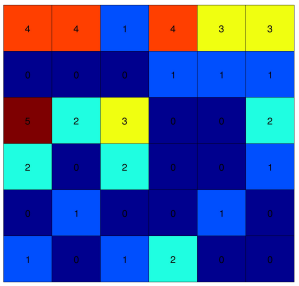


(e) Représentation des blocs de variables

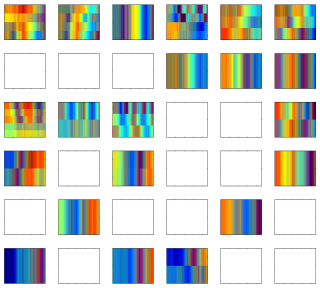
FIGURE 4.2 – Visualisation de la base de données isolet5 en utilisant BiTM. Chaque cellule des figures 4.2(c) et 4.2(d) indique une cellule de la carte.



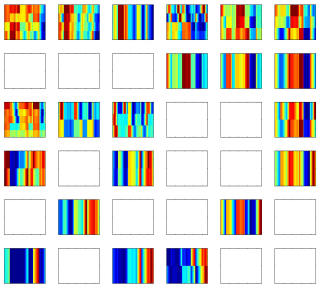
(a) La base de données organisée en fonction de l'ordre des observations et des variables de la classification croisée.



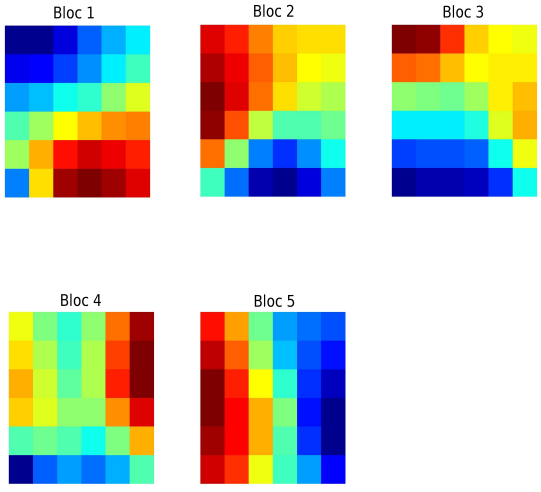
(b) Cardinalités de la carte.



(c) La carte BiTM. Représentation topologique des groupes de la carte BiTM.

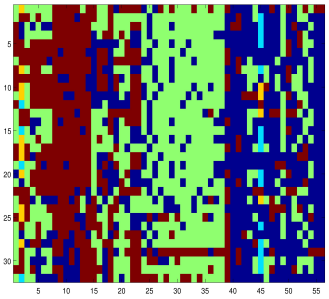


(d) Carte BiTM organisée selon les blocs de variables

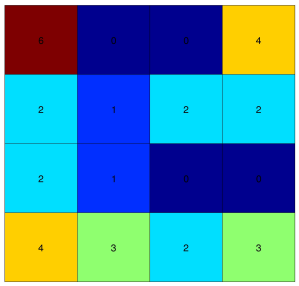


(e) Représentation des blocs de variables

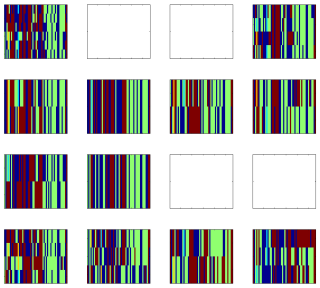
FIGURE 4.3 – Visualisation de la base de données Movement Libras en utilisant BiTM. Chaque cellule des figures 4.3(c) et 4.3(d) indique une cellule de la carte.



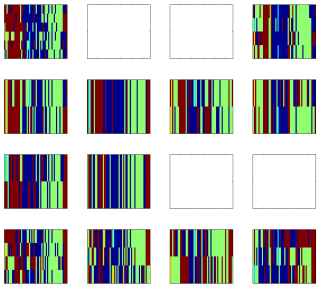
(a) La base de données organisée en fonction de l'ordre des observations et des variables de la classification croisée.



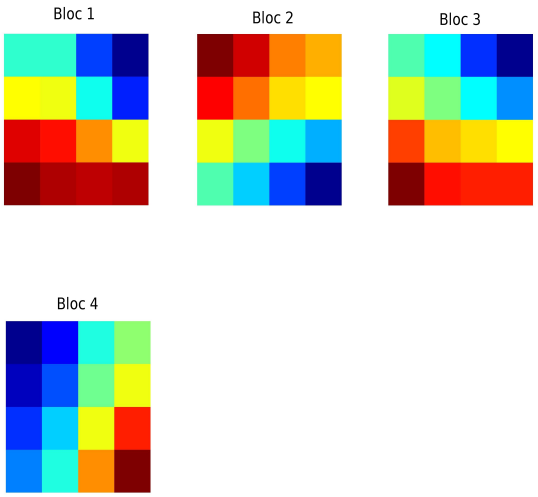
(b) Cardinalités de la carte.



(c) La carte BiTM. Représentation topologique des groupes de la carte BiTM.

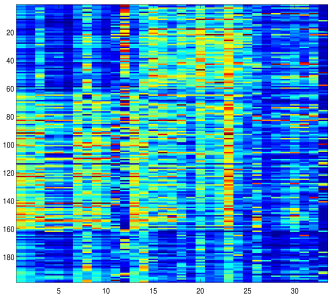


(d) Carte BiTM organisée selon les blocs de variables

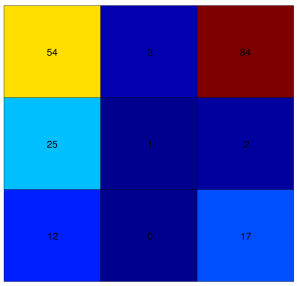


(e) Représentation des blocs de variables

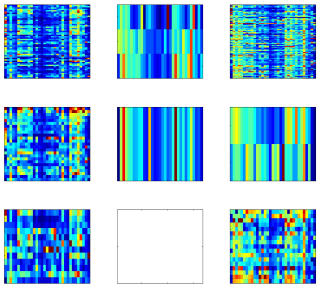
FIGURE 4.4 – Visualisation de la base de données Lung Cancer en utilisant BiTM. Chaque cellule des figures 4.4(c) et 4.4(d) indique une cellule de la carte.



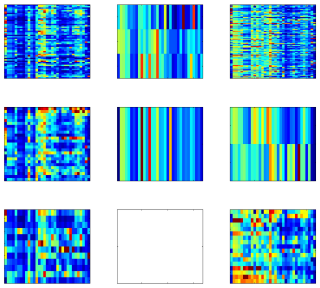
(a) La base de données organisée en fonction de l'ordre des observations et des variables de la classification croisée.



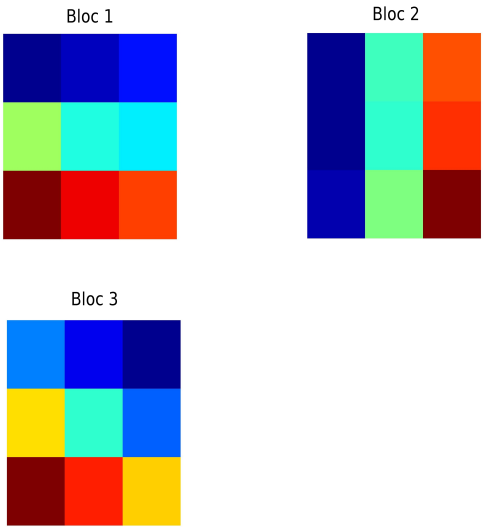
(b) Cardinalités de la carte.



(c) La carte BiTM. Représentation topologique des groupes de la carte BiTM.

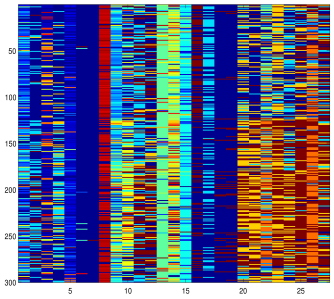


(d) Carte BiTM organisée selon les blocs de variables

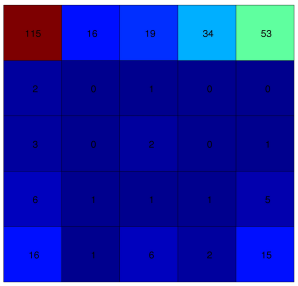


(e) Représentation des blocs de variables

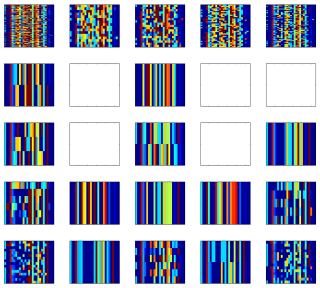
FIGURE 4.5 – Visualisation de la base de données Cancer Wpbc Ret en utilisant BiTM. Chaque cellule des figures 4.5(c) et 4.5(d) indique une cellule de la carte.



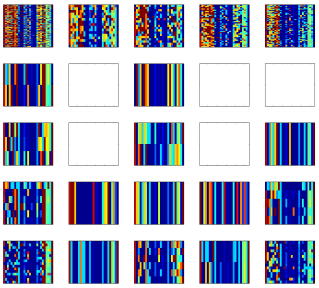
(a) La base de données organisée en fonction de l'ordre des observations et des variables de la classification croisée.



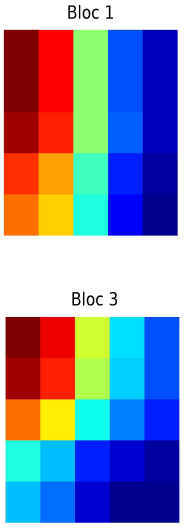
(b) Cardinalités de la carte.



(c) La carte BiTM. Représentation topologique des groupes de la carte BiTM.

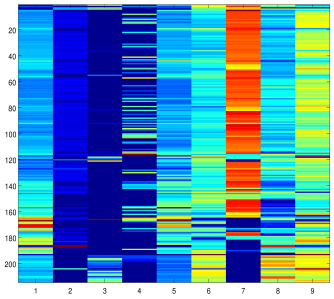


(d) Carte BiTM organisée selon les blocs de variables

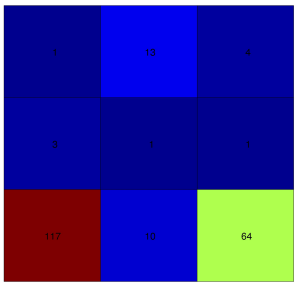


(e) Représentation des blocs de variables

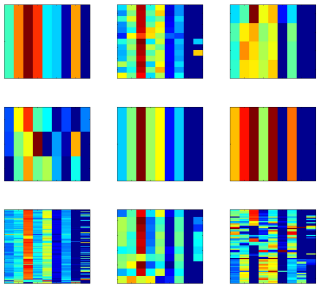
FIGURE 4.6 – Visualisation de la base de données HorseColic en utilisant BiTM. Chaque cellule des figures 4.6(c) et 4.6(d) indique une cellule de la carte.



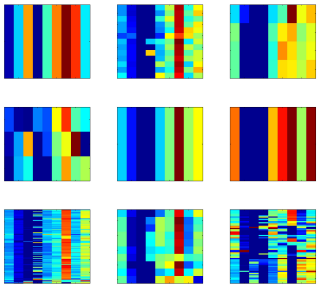
(a) La base de données organisée en fonction de l'ordre des observations et des variables de la classification croisée.



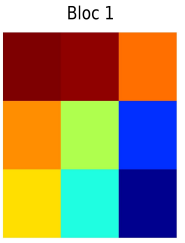
(b) Cardinalités de la carte.

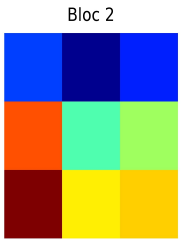


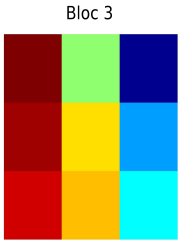
(c) La carte BiTM. Représentation topologique des groupes de la carte BiTM.



(d) Carte BiTM organisée selon les blocs de variables







(e) Représentation des blocs de variables

FIGURE 4.7 – Visualisation de la base de données glass en utilisant BiTM. Chaque cellule des figures 4.7(c) et 4.7(d) indique une cellule de la carte.

4.3 Conclusion

Nous avons constaté, après l'étude comparative avec des méthodes de partitionnement et de bi-partitionnement, que BiTM est une méthode de bi-partitionnement efficace. La principale nouveauté de BiTM est l'utilisation d'un modèle topologique pour organiser la matrice des données en blocs homogènes en prenant en compte simultanément les lignes et les colonnes. La série d'expériences que nous avons réalisée nous a permis de valider notre méthode et d'analyser ses performances à partir de nombreux critères. Ces résultats expérimentaux démontrent que notre algorithme identifie les bi-clusters et donne de bonnes performances par rapport à certains algorithmes de bi-partitionnement. Il offre aussi de nouvelles visualisations permettant de mieux comprendre la structure des données.

La tendance actuelle d'un accroissement fort de la taille des bases de données pose un défi sans précédent pour la fouille de données. Non seulement les bases de données s'agrandissent, mais de nouveaux types de données deviennent très répandus, tels que les flux de données sur le web, les données de puces à ADN génomique et les données relatives aux réseaux sociaux. Les chercheurs se sont rendu compte que la pondération de variables est un élément essentiel pour que la fouille de données atteigne ses objectifs [Weston 2000], [Guyon 2003], [Evgeniou 2004], [Tsai 2012]. Un nombre élevé de variables peut en effet s'avérer pénalisant pour un traitement pertinent et efficace des données, d'une part par les problèmes algorithmiques que cela peut entraîner (liés au coût calculatoire et à la capacité de stockage nécessaire), et d'autre part, la non pertinence, l'inutilité et/ou la redondance de certaines variables, perturbant ainsi le bon traitement des données.

Nous avons constaté à travers l'état de l'art réalisé sur les problèmes de bi-partitionnement qu'il existe un manque considérable sur le sujet de la pertinence de blocs de variables. Dans un cadre de bi-partitionnement, la notion de variable a un poids conséquent sur la modélisation du problème. C'est ainsi qu'il est souhaitable de connaître l'importance (pertinence) de chaque bloc de variables. Ceci permet de répondre à des applications à la fois importantes et intéressantes dans divers domaines. Nous citons entre autres la réduction de l'espace multidimensionnel en procédant par la sélection des blocs pertinents.

Chapitre 5

Contribution Anticipeo : estimation des intervalles de confiance et classification des produits

Sommaire

5.1 Introduction et problématique

Chaque mois, Anticipeo produit des projections (prévisions) de vente détaillées. Les modèles utilisés sont un ensemble de sous-modèles basés essentiellement sur des régressions linéaires [1] en appliquant des analyses sur les séries chronologiques.

À partir des historiques détaillés de ventes (appelées aussi "réalisations"), Anticipeo réalise des projections sur les 12 à 24 mois à venir, au même niveau de détail que les historiques. Ces projections sont élaborées par un moteur statistique qui est paramétré spécifiquement pour chaque société cliente. Ces projections sont ensuite contrôlées quant à leur pertinence (il s'agit essentiellement de repérer les événements exceptionnels ainsi que les données qui ne reflètent pas la réalité), afin que les projections délivrées aux managers et utilisateurs de la société cliente soient une bonne image des ventes qui devraient advenir, si les tendances passées continuaient à évoluer au même

1. Voir annexe B

rythme. Autrement dit, ces projections reflètent le futur le plus probable, s'il n'y a pas de discontinuité par rapport au passé. Bien évidemment, le futur n'est pas nécessairement la continuation des évolutions passées, notamment si des événements métiers impliquent des ruptures. Celles-ci peuvent provenir :

- Des plans d'action menés par l'entreprise cliente (plan de communication, actions de promotion des ventes, renouvellement de la gamme, etc.),
- De la concurrence, soit qui se crée, soit qui disparaît ou qui agit de façon différente par rapport au passé (si la concurrence évolue comme par le passé, les impacts auront déjà été intégrées dans les projections).

Ces facteurs sont du domaine de la connaissance métier de l'entreprise cliente : Anticipeo met à disposition de l'entreprise les outils qui lui permettent de transformer facilement et rapidement les projections en prévision. Pour schématiser, le principe d'Anticipeo est :

- Assurer la prise en charge de la partie mathématique par les équipes Anticipeo afin de gagner en précision et rapidité tout en diminuant les coûts de fabrication et de suivi des prévisionnels,
- Confier à l'entreprise la finalisation des prévisions pour intégrer son expertise métier.

L'estimation des intervalles de confiance permet à Anticipeo de donner des informations supplémentaires à ses clients sur le comportement de leurs produits. L'intervalle de confiance (IC) à $\alpha\%$ est un intervalle de valeurs qui a $\alpha\%$ de chances de contenir la vraie valeur du paramètre estimé. Il est possible de dire que l'IC représente la fourchette des valeurs à l'intérieur de laquelle nous sommes certains à $\alpha\%$ de trouver la vraie valeur recherchée. L'intervalle de confiance est donc l'ensemble des valeurs raisonnablement compatibles avec le résultat observé (l'estimation ponctuelle). Il donne une visualisation de l'incertitude de l'estimation.

Anticipeo produit chaque mois des projections, ce qui génère plusieurs valeurs d'écarts projections/réalisations que l'on note σ. Quel est le meilleur estimateur de σ en utilisant les écarts entre les différentes projections antérieures et les réalisations réelles ? Dans le cas où σ est une estimation de l'écart empirique dans un modèle de régression linéaire, quels sont les estimateurs des intervalles de confiance futurs ? Afin de répondre à cette problématique, nous nous sommes basées sur les résultats théoriques de la méthode de la régression linéaire simple et nous avons développé une approche empirique qui permet d'estimer les intervalles de confiance des projections établies par Anticipeo. Sachant que les écarts σ de nos modèles (variables indépendantes et identiquement distribuées) se comportent d'une manière paraboloïde dans le futur [McCullagh 2009].

Notre idée est basée sur le fait qu'une projection à long terme est moins

fiable que celle du court terme. À partir de ce résultat, nous avons attribué des pondérations (poids) par ordre décroissant chronologique en racine de temps ($\sqrt{t}$) sur chaque projection. Cela permet d'etimer σ dans le but de déduire les intervalles de confiance futurs.

5.2 Estimation des intervalles de confiance

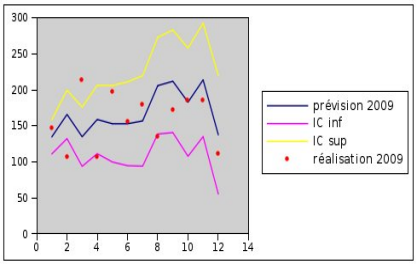
Définition : *l'écart-type empirique est la différence calculée entre les réalisations réelles et les projections élaborées par Anticipeo.*

Dans cette première partie, nous estimons les Intervalles de Confiance (IC) des projections de l'historique des bases de données d'Anticipeo où nous possédons l'information sur les réalisations réelles. Cette étape de l'estimation des IC ("marges d'erreur") consiste à calculer pour chaque produit à partir des écarts constatés entre projections et réalisations des mois -23 à -12 et estimation d'intervalles de confiance à 1, 2 ou 3 écarts-types entre les mois -11 et 0. Cette manière d'estimation permet de valider notre approche en mesurant la qualité des projections par le biais de la différence entre les réalisations et les projections.

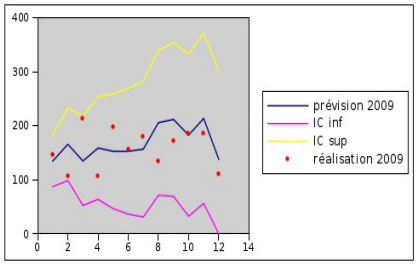
À partir des écarts quadratiques passés entre les projections antérieures et les réalisations en tenant compte du fait que les estimations des écarts grandissent proportionnellement à la racine carrée du temps, nous calculons pour chaque série un écart-type prévisionnel pour le mois donné ; cet écart-type sert à son tour pour estimer les intervalles de confiance pour le futur.

Les figures 5.1, 5.2, 5.3, 5.4, 5.5 et 5.6 représentent les résultats obtenus suivant notre procédure de calcul des IC appliqués sur 6 produits extraits d'une base de données client.

- Les points en rouge représentent les réalisations réelles des produits p_1, p_2, p_3, p_4, p_5 et p_6.
- La courbe en bleu représente les projections réalisées par l'outil Anticipeo sur les produits p_1, p_2, p_3, p_4, p_5 et p_6.
- Les courbes en jaune et en violet représentent les bornes sup et inf de l'intervalle de confiance associé à chaque produit (p_1, p_2, p_3, p_4, p_5 et p_6) selon notre procédure de calcul.

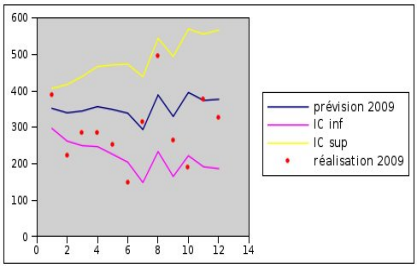


(a) Intervalle de confiance du produit p_1 avec $\sigma = 1$

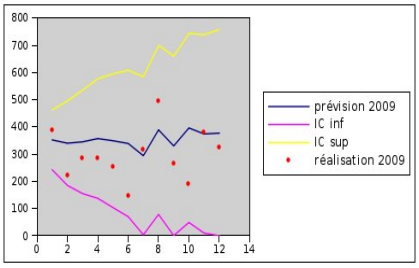


(b) Intervalle de confiance du produit p_1 avec $\sigma = 2$

FIGURE 5.1 – Intervalle de confiance du produit p_1 avec $\sigma = 1$ et avec $\sigma = 2$

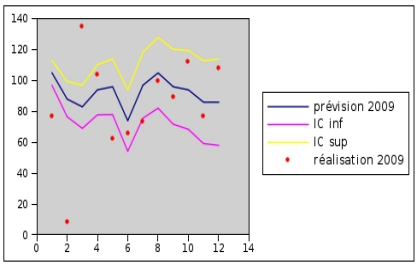


(a) Intervalle de confiance du produit p_2 avec $\sigma = 1$

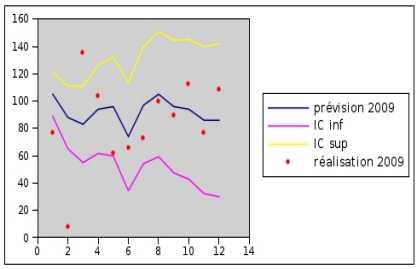


(b) Intervalle de confiance du produit p_2 avec $\sigma = 2$

FIGURE 5.2 – Intervalle de confiance du produit p_2 avec $\sigma = 1$ et avec $\sigma = 2$

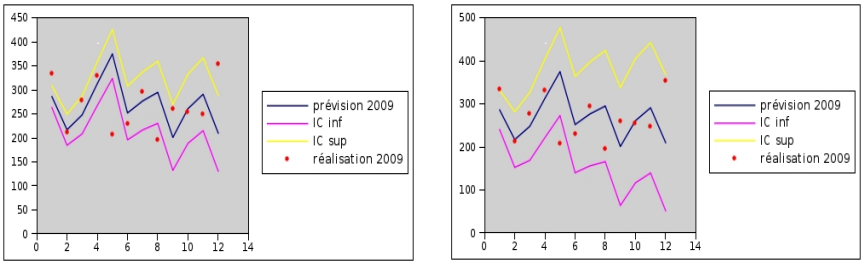


(a) Intervalle de confiance du produit p_3 avec $\sigma = 1$



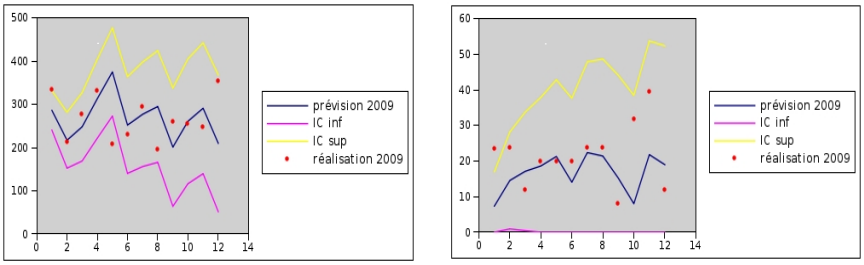
(b) Intervalle de confiance du produit p_3 avec $\sigma = 2$

FIGURE 5.3 – Intervalle de confiance du produit p_3 avec $\sigma = 1$ et avec $\sigma = 2$



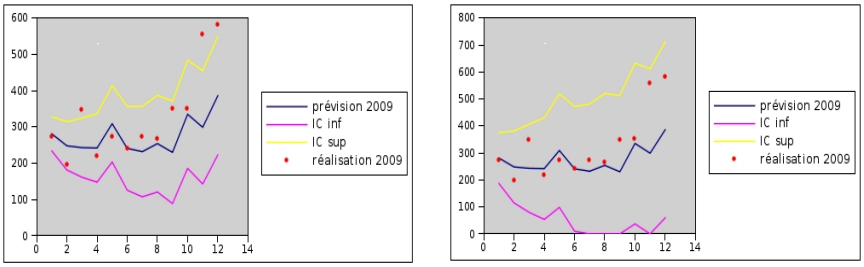
(a) Intervalle de confiance du produit p_4 avec $\sigma = 1$ (b) Intervalle de confiance du produit p_4 avec $\sigma = 2$

FIGURE 5.4 – Intervalle de confiance du produit p_4 avec $\sigma = 1$ et avec $\sigma = 2$



(a) Intervalle de confiance du produit p_5 avec $\sigma = 1$ (b) Intervalle de confiance du produit p_5 avec $\sigma = 2$

FIGURE 5.5 – Intervalle de confiance du produit p_5 avec $\sigma = 1$ et avec $\sigma = 2$



(a) Intervalle de confiance du produit p_6 avec $\sigma = 1$ (b) Intervalle de confiance du produit p_6 avec $\sigma = 2$

FIGURE 5.6 – Intervalle de confiance du produit p_6 avec $\sigma = 1$ et avec $\sigma = 2$

Nous remarquons à partir de ces produits qu'on arrive à obtenir des résultats intéressants quant à l'estimation des intervalles de confiance. En effet,

dans la plupart des cas, on a réussi à obtenir les réalisations à l'intérieur de l'intervalle de confiance selon le nombre d'écarts avec lequel on a estimé les bornes inférieures et supérieures des IC. Dans certains cas, il existe un écart significatif entre la réalisation et la projection. Comme notre IC se base essentiellement sur les projections, alors il est difficile d'estimer les bornes inférieures et supérieures de ce genre de réalisations. C'est le cas des réalisations des mois 2 et 3 du produit p_3.

Jusqu'à cette étape, nous avons travaillé dans un cadre totalement supervisé. Cela veut dire qu'on pouvait estimer la qualité des projections élaborées en les confrontant aux réalisations réelles observées. L'objectif final étant le calcul des intervalles de confiance pour les projections, c'est-à-dire des mois 1 à 12. Les différentes étapes qui permettent de calculer les intervalles de confiance des mois 1 à 12 restent les mêmes que celles de l'étape 1, sauf que l'estimation des écarts-types empiriques s'effectue sur les mois -11 à 0 et que les intervalles de confiance sont établis pour les mois 1 à 12.

Après avoir calculé les intervalles de confiance pour les mois 1 à 12, nous ajustons les projections sur les mois 12 à 24 à venir en nous basant sur la tendance et la saisonnalité des produits dans le passé.

5.3 Calcul du "taux de confiance réel" et résultats obtenus

Définition : *le "taux de confiance réel" est le nombre de réalisations dans l'intervalle de confiance sur le nombre de réalisations total.*

Toute la difficulté est de savoir si on est bien sur une loi normale et si donc cette estimation empirique de σ est sans biais. Nous testons toujours sur le passé le nombre d'écarts-types empiriques pour obtenir un niveau de confiance donné.

C'est ainsi que nous avons développé une procédure empirique qui permet de calculer le "taux de confiance réel" (nombre de réalisations dans l'intervalle sur nombre total) pour les mois -11 à 0 en utilisant les intervalles de confiance des mois -11 à 0.

Soit la variable "out" qui représente le nombre de réalisations sortantes de l'intervalle de confiance. "out" varie entre 0 et 12 ("out"= 0...12). "out-plus" est le nombre de réalisations qui sont supérieures à la borne sup de l'intervalle de confiance. "out-moins" est le nombre de réalisations qui sont inférieures à la borne inf de l'intervalle de confiance. Le niveau de confiance de nos IC est calculé comme suit :

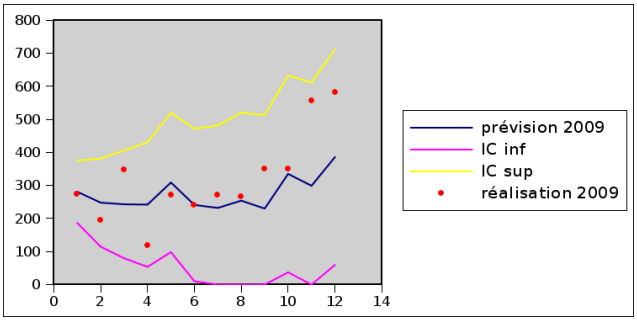
– Si on a 0 réalisation à l'extérieur de l'IC, c'est-à-dire que le couple (out-

plus,out-moins)=(0,0), alors le niveau de confiance est de $1-(0/12)=1=100\%$.

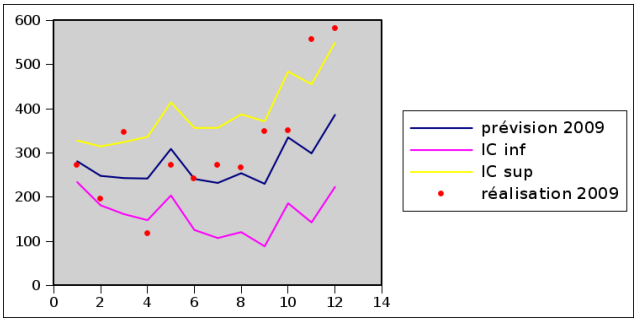
- Si on a 1 réalisation à l'extérieur de l'IC, c'est-à-dire que le couple (out-plus,out-moins)=(1,0) ou (0,1), alors le niveau de confiance est $1-(1/12)=92\%$.
- Si on a 2 réalisations à l'extérieur de l'IC, c'est-à-dire que le couple (out-plus,out-moins)=(2,0) ou (0,2) ou (1,1), alors le niveau de confiance est $1-(2/12)=83\%$.
- Si on a 3 réalisations à l'extérieur de l'IC, c'est-à-dire que le couple (out-plus,out-moins)=(3,0) ou (0,3) ou (2,1) ou (1,2), alors le niveau de confiance est $1-(3/12)=75\%$.

Exemple d'application : prenons un produit qui a 100 % de confiance, cela veut dire qu'il n'y a aucune réalisation à l'extérieur de l'IC. On souhaite réduire le niveau de confiance de ce produit à 75 % de confiance afin d'obtenir des intervalles de confiance de longueur réduite (2 σ par défaut). Nous appliquons donc notre procédure afin d'obtenir le nombre de sigma optimal ($nb\sigma$) qui correspond au niveau de confiance 75 %.

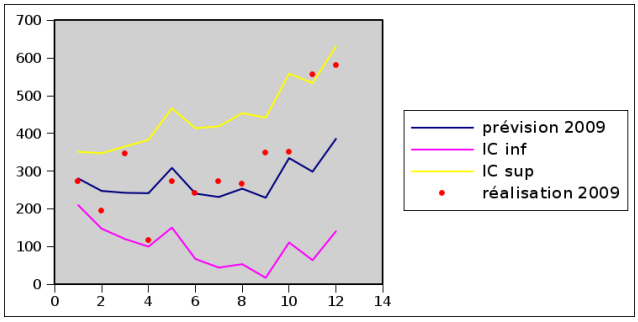
- Série initiale (par défaut $nb\sigma = 2$).



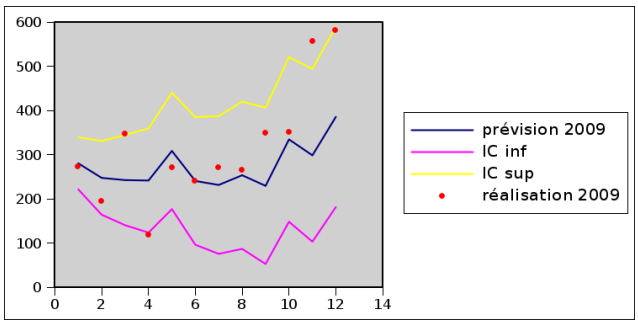
- Itération 1 :



- Itération 2 :



- Itération 3 :



La procédure se termine ainsi, car nous avons exactement 3 réalisations à l'extérieur de l'IC, qui correspond à un niveau de confiance de 75 %, le nombre de sigma optimal dans ce cas est $nb\sigma = 1.25$.

5.4 Classification des produits

Afin d'effectuer une meilleure analyse sur les bases de données d'Anticipeo, nous effectuons un clustering sur ces bases qui représentent les ventes des produits d'un de ses clients. L'algorithme de clustering utilisé est GOF-SOM (voir chapitre 3). Le choix d'appliquer le modèle GOF-SOM comme algorithme d'apprentissage revient au fait de donner une dimension visuelle aux données afin de mieux comprendre leur structure. Nous visons avec cette classification à regrouper les produits dans différentes classes distinctes selon leur l'homogénéité à partir de leurs caractéristiques statistiques, telles que :

- Trend des ventes (ventes de -11 à 0 sur ventes de -23 à -12),
- Indice de corrélation des coefficients saisonniers,
- Écart-type empirique en % (écart-type des intervalles de confiance divisé par ventes moyennes mensuelles des 12 derniers mois),
- Nombre de sigma final ($nb\sigma$ optimal).

La base de données initiale contient 5897 produits. Après avoir effectué un prétraitement sur cette base et en éliminant les produits qui ont plus de 24 mois de vente dans l'historique, la base de données sur laquelle nous réaliserons le clustering contient uniquement 807 produits.

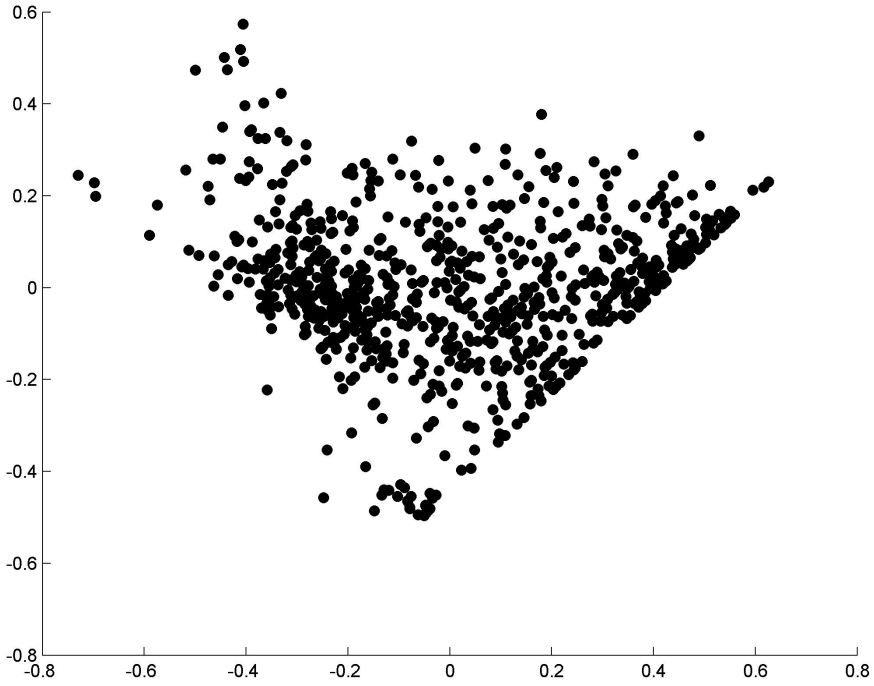


FIGURE 5.7 – Projection des données d'Anticiepo en 2 dimensions en utilisant l'ACP

La figue 5.7 représente une projection en 2 dimensions en utilisant une ACP. Nous remarquons à travers cette projection que quelques zones sont de forte densité. Nous remarquons aussi la présence de quelques données isolées dans les zones frontières, et d'un groupe de données isolé et relativement dense (au centre en bas de la figure).

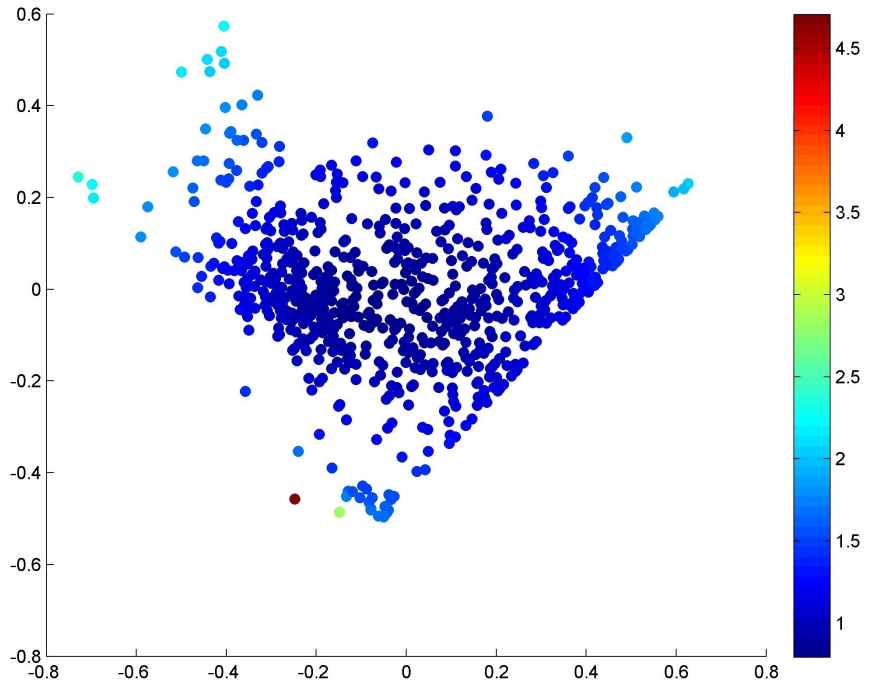


FIGURE 5.8 – LOF appliqué sur les données d'Anticiepo

Nous avons appliqué l'algorithme de LOF (voir le chapitre 2, algorithme 8) sur les données. Les résultats obtenus sont illustés dans la figure 5.8. Cet algorithme a détecté 2 outliers, le point en rouge (que nous appelons $outlier_1$) a une valeur de LOF égale à 4.71, et le point en vert (que nous appelons $outlier_2$) a une valeur de LOF égale à 2.86. En analysant ces deux produits, nous avons remarqué que $outlier_1$ a un écart-type empirique égale à 0.008 et $outlier_2$ 0.02. Ceci veut dire que l'intervalle de confiance a une longueur quasiment nulle. Nous avons analysé les autres variables de ces produits et nous avons constaté, que pour l'$outlier_2$, le trend des ventes est égal à 2398.14 et le nombre de sigma final est égal à 1402,84. Pour *l'*$outlier_2$, le trend des ventes est égale à 1312.54 et le nombre de sigma final est égal à 738.95. Ces valeurs pour les produits de cette base de données sont aberrantes.

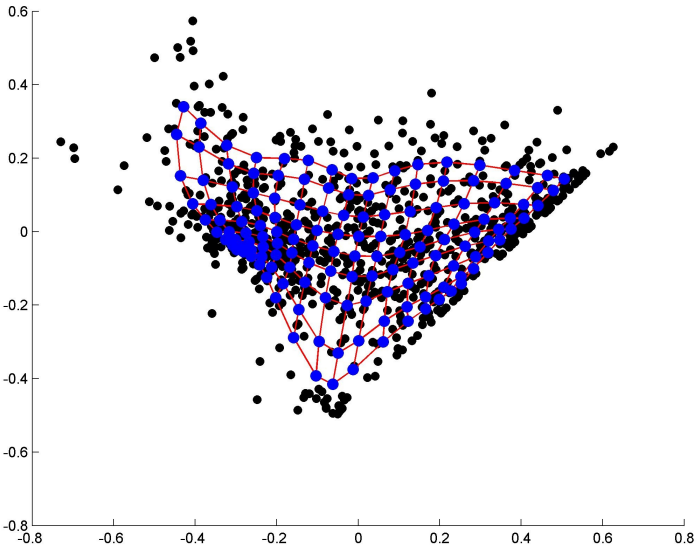


FIGURE 5.9 – Classification des produits : GOF-SOM appliqué sur les données d'Anticiepo

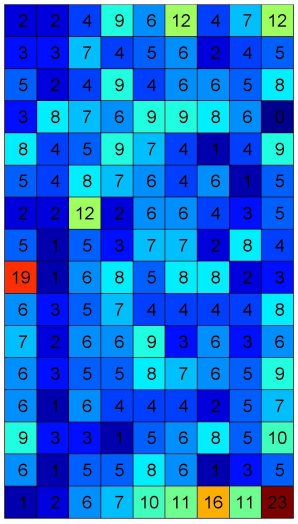


FIGURE 5.10 – Cardinalité de la carte GOF-SOM appliquée sur les données d'Anticiepo

Nous avons appliqué l'algorithme GOF-SOM sur cette base de données. Nous remarquons à partir de la figure 5.9 que la carte GOF-SOM épouse le nuage de points. Les cardinalités de la carte obtenue sont représentées dans la figure 5.10.

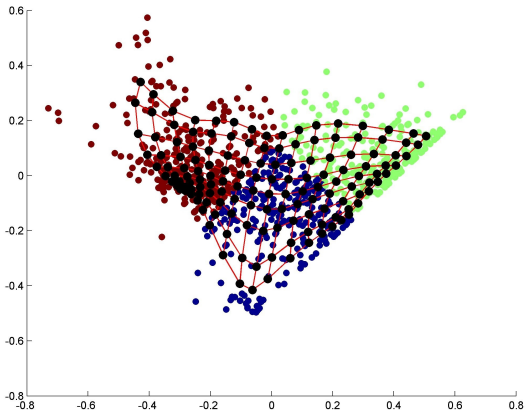


FIGURE 5.11 – Segmentation de la carte GOF-SOM appliquée sur les données d'Anticiepo. Les couleurs représentent les classes obtenues.

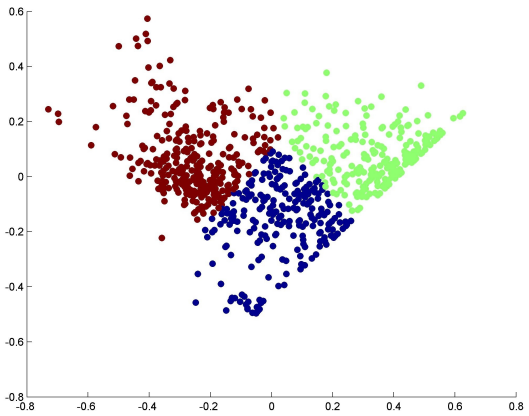


FIGURE 5.12 – Les classes obtenues en segmentant la carte GOF-SOM appliquée sur les données d'Anticiepo

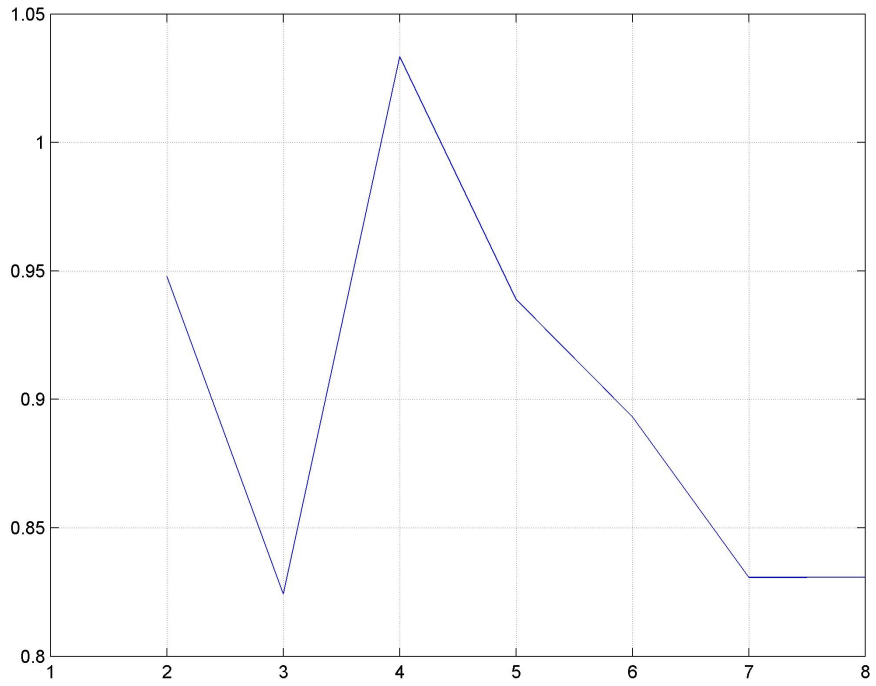


FIGURE 5.13 – Indice Davies-Bouldin calculé sur les référents de la carte GOF-SOM appliqué sur les données d'Anticiepo. Sur l'axe des abscisses, nous représentons le nombre de classes (k), et sur l'axe des ordonnées, nous représentons les valeurs de l'indice de Davies-Bouldin.

La figure 5.11 représente la carte GOF-SOM segmentée. Nous avons d'abord appliqué l'algorithme GOF-SOM sur les données. Ensuite, nous avons appliqué l'algorithme K-means standard sur les référents de la carte. Nous avons obtenu 3 classes de produits en utilisant l'indice Davies-Bouldin représenté dans la figure 5.13. La figure 5.12 représente les classes obtenues en segmentant la carte GOF-SOM appliquée sur les données d'Anticiepo.

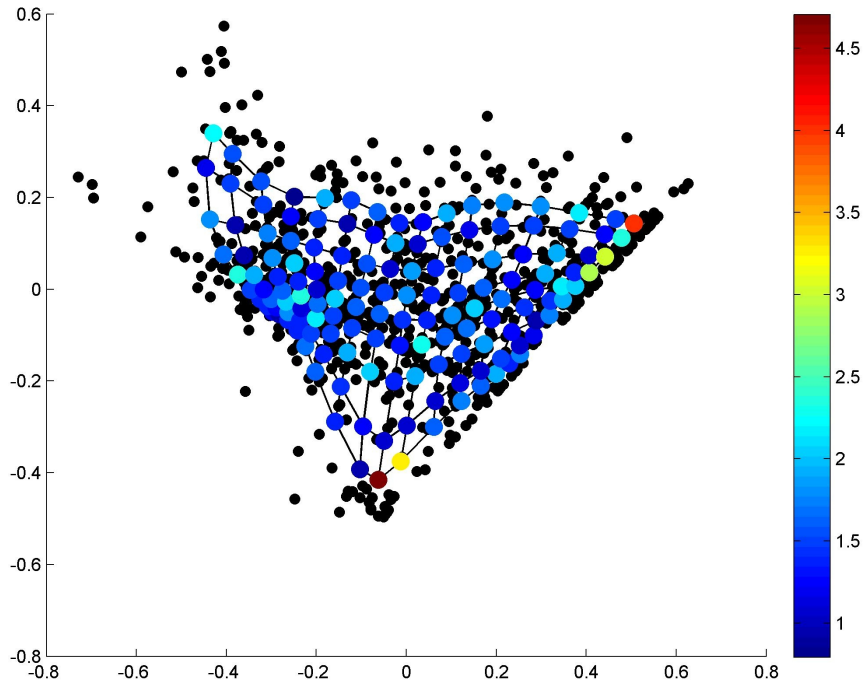


FIGURE 5.14 – Détection de groupes-outliers : GOF-SOM appliqué sur les données d'Anticiepo

En analysant le score GOF appliqué à ces données, nous avons remarqué que les produits qui forment ce groupe ont tous un nombre de sigma final très élevé (> 20). Ceci veut dire que, pour atteindre un niveau de confiance précis, il faut un nombre de sigma final très important. Ceci représente, donc, une anomalie par rapport à cette base de données.

L'analyse réalisée sur les données d'Anticipeo permet une meilleur compréhension des données et fournit des résultats intéressants sur la détection d'outliers et de groupes-outliers, la classification et la visualisation des données.

5.5 Conclusion

Nous avons présenté dans ce chapitre les différentes contributions liées aux données d'Anticipeo. En premier lieu, nous avons développé une approche qui permet d'estimer les intervalles de confiance et le calcul du "taux de confiance réel". Ensuite, nous avons appliqué les différentes approches développées dans

cette thèse (classification des produits, détection des outliers et des groupes-outliers) aux données d'Anticipeo. Les résultats obtenus sont intéressants et prometteurs.

Conclusion générale et perspectives

Conclusion générale

Nous avons présenté d'abord dans cette thèse un panorama bibliographique sur les méthodes d'apprentissage non supervisé (clustering et bi-clustering) ainsi que les différentes méthodes existant dans la littérature sur la détection d'outliers et de la détection de nouveautés. Nous avons ensuite présenté nos différentes contributions.

La première contribution de cette thèse concerne la détection de groupes-outliers et la détection de nouveautés. Nous avons présenté un nouveau paramètre GOF basé sur les densités locales des clusters. Ce paramètre a été intégré aux cartes auto-organisatrices. Une série d'expériences a été réalisée pour valider la méthode proposée. Ceci nous a permis de mieux évaluer notre approche, qui s'est avérée prometteuse comme solution au problème de détection de groupes-outliers. Le score GOF est ensuite utilisé comme classifieur pour le problème de détection de nouveautés. Les résultats obtenus en comparant notre approche avec des approches classiques de détection de nouveautés étaient satisfaisants.

La deuxième contribution de cette thèse est la proposition d'une nouvelle approche de bi-partitionnement topologique. La principale nouveauté du BiTM est l'utilisation d'un modèle topologique pour organiser la matrice des données en blocs homogènes, tout en prenant en compte simultanément les lignes et les colonnes ainsi que la visualisation des bi-clusters organisés dans une carte topologique. La série d'expériences que nous avons réalisée nous a permis de valider notre méthode et d'analyser ses performances à partir de nombreux critères. Ces résultats expérimentaux démontrent que notre algorithme identifie les bi-clusters et a de bonnes performances par rapport à certains algorithmes de classification croisée. Nous avons constaté, après l'étude comparative avec des méthodes de clustering et de bi-clustering, que BiTM est une méthode de bi-partitionnement efficace. Nous avons aussi montré qu'il existe un lien fort avec la méthode Croeuc.

La dernière contribution de cette thèse concerne l'estimation des intervalles de confiance et la classification des produits Anticipeo. Nous avons mis en place une méthode qui permet d'estimer les intervalles de confiance et le calcul du taux de confiance "réel". Enfin, nous avons appliqué les différentes approches développées dans cette thèse (classification des produits,

détection des outliers, groupes-outliers et nouveautés et bi-partitionnement des données) aux données d'Anticipeo. Les résultats obtenus sont intéressants et performants. Aussi, ils ouvrent de nouvelles perspectives pour Anticipeo au développement de nouvelles offres commerciales.

Perspectives

Nombreuses sont les perspectives de recherche qu'offre cette thèse. En effet, adapter l'approche GOF-SOM au flux de données est une perceptive incontournable, car les groupes que nous considérons comme "groupes-outliers" peuvent être classés "clusters-normaux" si un grand nombre de données sont affectées aux "groupes-outliers". L'enjeu est la détermination, ou du moins l'estimation du seuil de passage d'un groupe-outlier à un cluster-normal.

Dans cette thèse, nous avons testé nos différentes approches sur des bases de données de tailles réduites pour mieux étudier les algorithmes et visualiser les données ainsi que les groupes-outliers. Il est évident que le but de notre méthode est non pas de l'appliquer sur des bases réduites, mais de l'utiliser dans des bases de données multidimensionnelles, complexes, déséquilibrées et de tailles importantes (big data). Cela constitue une autre perspective de nos travaux. Nous essayons aussi d'améliorer davantage les performances de calcul de GOF en testant de nouvelles fonctions de densité. Il serait judicieux d'intégrer GOF aux données binaires et mixtes et de tester un autre seuil pour la détection de nouveautés afin d'évaluer au mieux notre approche.

Nous proposons aussi de réaliser des graphiques de types boîte à moustache, cartes topologiques et histogrammes, afin de permettre une meilleure compréhension et interprétation des résultats aux utilisateurs concernant les applications aux données Anticipeo.

Il y a lieu de rappeler que l'utilisateur qui veut couvrir tous les aspects existants d'un problème particulier et obtenir une connaissance compréhensible doit considérer un grand nombre de variables. Or, parmi ces variables, certaines sont inutiles. En effet, il est souvent difficile voire impossible de discerner les variables pertinentes des variables non pertinentes, ce qui pousse l'utilisateur à s'emparer de toutes les variables disponibles. De plus, les sources de données peuvent être multiples, et la fusion des données issues de chacune de ces sources conduit à la création d'un ensemble contenant des variables inutiles et redondantes. La solution que l'on peut apporter à cette difficulté est la pondération de "blocs de variables". Ces blocs sont obtenus en appliquant l'approche BiTM que nous avons proposé, dans cette thèse. La pondération/sélection de blocs de variables est un processus permettant l'élimination des blocs de variables inutiles et/ou redondantes, et l'élimination du bruit

pouvant être généré par certaines variables. Ceci représente un des axes importants des perspectives de cette thèse que nous avons commencé à explorer. La formulation du modèle ainsi que les résultats préliminaires obtenus sont détaillés dans l'annexe A.

Annexe A

Pondération de blocs de variables

Nous disposons de quelques résultats préliminaires sur la pondération de blocs de variables. Le modèle FBR_BiTM (Feature Block Relevance using BiTM)) est basé sur l'approche BiTM en attribuant à chaque bloc de variables un nouveau score de pertinence nommé **fbr**. Ce score est calculé durant la phase d'apprentissage et représente la pertinence de chaque bloc de variables. Nous proposons de minimiser la nouvelle fonction de coût suivante :

$$\begin{aligned}\tilde{R}(\phi_w, \phi_z, G, FBR) &= \sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{\mathbf{x}_i \in P_k}\sum_{\mathbf{x}^j \in Q_l}\sum_{r=1}^{L}\mathcal{K}^T(\delta(r,k)) \\ &\times \ (fbr_r^l \times x_i^j - g_r^l)^2\end{aligned}$$

Où :

$G = \{\mathbf{g_1}, \ldots, \mathbf{g_k}\}$ désigne l'ensemble des prototypes.
$\mathbf{FBR} = \{\mathbf{fbr_1}, \ldots, \mathbf{fbr_k}\}$ représente l'ensemble des scores de pertinence et qui ont la même dimension des prototypes.
ϕ_z est la fonction d'affectation des lignes.
ϕ_w est la fonction d'affectation des colonnes.
$\mathcal{K}^T(\delta(r,k))$ est la fonction de voisinage.
T représente la fonction contrôlant le rayon du voisinage.
En pratique, nous utilisons la fonction de voisinage suivante : $\mathcal{K}^T(\delta(c,r)) = \exp\left(\frac{-\delta(c,r)}{T}\right)$.

Affectation des observations : chaque observation $\mathbf{x}_i$ est affectée au prototype $\mathbf{g}_k$ le plus proche en utilisant la fonction d'affectation :

$$\phi_z(\mathbf{x}_i) = \arg\min_c \sum_{j=1}^{d}\sum_{l=1}^{m}\sum_{r=1}^{K} w_{jl} \times \mathcal{K}^T(\delta(r,c)) \times (fbr_r^l \times x_i^j - g_r^l)^2$$

Affectation des variables : chaque variable $\mathbf{x}^j$ est affectée au prototype $\mathbf{g}_k^l$ le plus proche en utilisant la fonction d'affectation :

$$\phi_w(\mathbf{x}^j) = \arg\min_l \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{r=1}^{K} z_{ik} \times \mathcal{K}^T(\delta(r,k)) \times (fbr_r^l \times x_i^j - g_r^l)^2$$

Mise à jour des prototypes : les vecteurs des prototypes sont mis à jour suivant la formule ci-dessous :

$$g_k^l = \frac{\sum_{i=1}^{N}\sum_{j=1}^{d} \mathcal{K}^T(\delta(k,\phi_z(\mathbf{x}_i))) \times w_{jl} \times x_i^j \times fbr_r^{l/j\in Q_l}}{\sum_{i=1}^{N}\sum_{j=1}^{d} \mathcal{K}^T(\delta(k,\phi_z(\mathbf{x}_i))) \times w_{jl}}$$

Mise à jour des scores de pertinence : les vecteurs des scores de pertinence sont mis à jour suivant la formule ci-dessous :

$$fbr_r^l = \frac{\sum_{i=1}^{N}\sum_{j=1}^{d} \mathcal{K}^T(\delta(k,\phi_z(\mathbf{x}_i))) \times w_{jl} \times x_i^j \times g_k^{l/l<d}}{\sum_{i=1}^{N}\sum_{j=1}^{d} \mathcal{K}^T(\delta(k,\phi_z(\mathbf{x}_i))) \times w_{jl} \times (x_i^j)^2}$$

Afin de vérifier que le nouveau terme introduit à la fonction de coût ne perturbe pas le bi-clustering, nous avons comparé le modèle FBR-BiTM avec le modèle BiTM en utilisant les mesures de performance définies dans la section 4.2.1 du chapitre 4.

dataset	FBR_BiTM	BiTM
isolet5	0.441	0.316
MovementLibras	0.644	0.712
Breast	0.968	0.978
SonarMines	0.730	0.769
LungCancer	0.843	1
Spectf 1	0.727	0.759
Cancer Wpbc Ret	0.772	0.787
HorseColic	0.673	0.719
Heart	0.814	0.883
glass	0.439	0.618

TABLE A.1 – Indice de pureté (ACC) obtenu avec FBR_BiTM et BiTM.

dataset	FBR_BiTM	BiTM
isolet5	0.858	0.926
MovementLibras	0.933	0.937
Breast	0.790	0.687
SonarMines	0.524	0.508
LungCancer	0.687	0.459
Spectf 1	0.416	0.418
Cancer Wpbc Ret	0.435	0.435
HorseColic	0.474	0.472
Heart	0.615	0.56
glass	0.607	0.653

TABLE A.2 – Indice de rand obtenu avec FBR_BiTM et BiTM.

dataset	FBR_BiTM	BiTM
isolet5	0.525	0.439
MovementLibras	0.782	0.811
Breast	0.589	0.53
SonarMines	0.120	0.158
LungCancer	0.263	0.461
Spectf 1	0.110	0.1449
Cancer Wpbc Ret	0.030	0.081
HorseColic	0.029	0.06
Heart	0.224	0.247
glass	0.193	0.125

TABLE A.3 – Indice de NMI obtenu avec FBR_BiTM et BiTM.

Nous remarquons, d'après les tableaux A.1, A.2 et A.3, que FBR_BiTM et BiTM donnent presque les mêmes résultats dans la plupart des bases de données. Donc, l'introduction de FBR dans la fonction de coût ne perturbe pas le bi-partitionnement.

Annexe B

Généralités sur la régression

Sommaire

B.1 La régression linéaire simple

La régression linéaire simple est l'une des notions basiques de la statistique et de l'analyse des données. Généralement, le problème consiste à décrire la dépendance entre deux variables aléatoires X et Y. Les mécanismes probabilistes qui se cachent derrière cette dépendance pourront être compliqués, et c'est pourquoi on se contente d'une approximation de Y en fonction de X [Manski 1989].

$$Y \approx f(X)$$

La fonction f est appelée "régression" et le problème consiste à reconstruire (estimer) f à partir de X et Y. Très souvent, cette estimation a pour but une prévision. x est la variable indépendante ou explicative. Les valeurs de x sont fixées par l'expérimentateur et sont supposées connues sans erreur. y est la variable dépendante ou expliquée. Les valeurs de y sont entachées d'une erreur de mesure. L'un des buts de la régression sera précisément d'estimer cette erreur et de chercher une relation de cette forme [Manski 1989] :

$$y = b_0 + b_1 x$$

C'est l'équation d'une droite, d'où le terme de "régression linéaire".

Les points expérimentaux ne se situent pas exactement sur la droite ; il faut donc trouver l'équation de la droite qui passe le plus près possible de ces points. On doit donc estimer les coefficients b_0 et b_1 qui réalisent le meilleur ajustement.

Estimation par les moindres carrées

La méthode des moindres carrés consiste à chercher les valeurs des paramètres b_0 et b_1 qui rendent minimale la somme des carrés des écarts résiduelle (SSr : sum of squared residuals) entre les valeurs observées y_k et les valeurs calculées de y [Cornillon 2007] :

$$SS_r = \sum_{k=1}^{n} (y_k - \hat{y}_k)^2 \tag{B.1}$$

Où n est le nombre de points et :

$$\hat{y} = b_0 + b_1 x \tag{B.2}$$

D'où :

$$SS_r = \sum_{k=1}^{n} (y_k - b_0 - b_1 x_k)^2 = \Phi(b_0, b_1) \tag{B.3}$$

Cette relation fait apparaître la somme des carrés des écarts comme une fonction des paramètres b_0 et b_1. Lorsque cette fonction est minimale, les dérivées par rapport à ces paramètres s'annulent :

$$\begin{cases} \frac{\partial \Phi}{\partial b_0}(b_0, b_1) = -2 \sum_{k=1}^{n} (y_k - b_0 - b_1 x_k) \\ \\ \frac{\partial \Phi}{\partial b_1}(b_0, b_1) = -2 \sum_{k=1}^{n} x_k (y_k - b_0 - b_1 x_k) \end{cases} \tag{B.4}$$

Soit :

$$\begin{cases} n b_0 + b_1 \sum_{k=1}^{n} x_k = \sum_{k=1}^{n} y_k \\ \\ b_0 \sum_{k=1}^{n} x_k + b_1 \sum_{k=1}^{n} x_k^2 = \sum_{k=1}^{n} x_k y_k \end{cases} \tag{B.5}$$

Le système B.5 est dit "système des équations normales". Il admet pour solutions :

$$\begin{cases} b_0 = \frac{\sum_{k=1}^{n} x_k^2 \sum_{k=1}^{n} y_k - \sum_{k=1}^{n} x_k \sum_{k=1}^{n} x_k y_k}{n \sum_{k=1}^{n} x_k^2 - \left(\sum_{k=1}^{n} x_k \right)^2} \\ \\ b_1 = \frac{n \sum_{k=1}^{n} x_k y_k - \sum_{k=1}^{n} x_k \sum_{k=1}^{n} y_k}{n \sum_{k=1}^{n} x_k^2 - \left(\sum_{k=1}^{n} x_k \right)^2} \end{cases} \tag{B.6}$$

Coefficients de détermination et de corrélation

Les auteurs de [Cameron 1997] montrent que :

$$\sum_{k=1}^{n}(y_k - \bar{y}_k)^2 = \sum_{k=1}^{n}(\hat{y}_k - \bar{y}_k)^2 + \sum_{k=1}^{n}(y_k - \hat{y}_k)^2 \tag{B.7}$$

$$SS_t = SS_e + SS_r$$

C'est l'équation d'analyse de la variance.

- SSt est la somme des carrés des écarts totaux. Elle traduit la dispersion des valeurs observées de y par rapport à la moyenne.
- SSe est la somme des carrés des écarts expliqués. Elle traduit la dispersion des valeurs calculées de y par rapport à la moyenne.

Si l'équation de la droite représente correctement les valeurs expérimentales, alors :

$$SS_e \to SS_t \iff r^2 = \frac{SS_e}{SS_t} \to 1 \tag{B.8}$$

r^2 est le "coefficient de détermination". Il représente la part des variations de y qui est "expliquée" par x. r est le coefficient de corrélation. Il est affecté du signe + ou − selon que la pente de la droite (b_1) est positive ou négative. r est toujours compris entre -1 et 1.

Variance expliquée, variance résiduelle

La variance expliquée est représentée par : $Ve = \frac{SSe}{p-1}$. Cependant, la variance résiduelle est représentée par : $Vr = \frac{SSr}{n-p}$ [Cox 2006].
Où p désigne le nombre de paramètres du modèle ($p = 2$ pour une droite). Si l'équation de la droite représente correctement les valeurs expérimentales, SSr doit tendre vers 0 et le rapport $F = \frac{Ve}{Vr}$ doit tendre vers l'infini.

Formulation matricielle de la régression linéaire

Le système des équations normales B.5 s'écrit sous forme matricielle [Meyer 2011] :

$$\begin{pmatrix} n & \sum x \\ \sum x & \sum x^2 \end{pmatrix} \cdot \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} \sum y \\ \sum xy \end{pmatrix} \tag{B.9}$$

$$A \quad . \quad B \quad = \quad C$$

A est la matrice du système, B le vecteur des paramètres et C le vecteur des termes constants. La matrice U et le vecteur Y sont introduits à la formulation ci-dessus telle que :

$$U = \begin{pmatrix} 1x_1 \\ 1x_2 \\ \dots \\ 1x_n \end{pmatrix} \qquad Y = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix} \tag{B.10}$$

$$A = U^T U \qquad C = U^T Y \tag{B.11}$$

Où le symbole T désigne la transposition matricielle (échange des lignes et des colonnes).

$$B = A^{-1}C = (U^T U)^{-1}(U^T Y) \tag{B.12}$$

La matrice inverse A^{-1} est donnée par :

$$A^{-1} = \frac{1}{det\ A} \begin{pmatrix} \sum x^2 & -\sum x \\ -\sum x & n \end{pmatrix} \tag{B.13}$$

$$det\ A = n\sum x^2 - \left(\sum x\right)^2$$

Où $det\ A$ est le déterminant de la matrice.

$$A^{-1} = A^{-1}C = \frac{1}{det\ A} \begin{pmatrix} \sum x^2 & -\sum x \\ -\sum x & n \end{pmatrix} . \begin{pmatrix} \sum y \\ \sum xy \end{pmatrix} \tag{B.14}$$

Soit :

$$B = \begin{pmatrix} \frac{\sum x^2 \sum y - \sum x \sum xy}{n\sum x^2 - (\sum y)^2} \\ \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum y)^2} \end{pmatrix} \tag{B.15}$$

La matrice de variance-covariance V est : $V = V_r A^{-1}$, où V_r est la variance résiduelle. Les termes diagonaux de V donnent les variances des paramètres, c'est-à-dire les carrés des écarts-types s_0 et s_1. Il vient donc :

$$s_0^2 = V_{00} = V_r \frac{\sum x^2}{n\sum x^2 - (\sum y)^2} \tag{B.16}$$

$$s_1^2 = V_{11} = V_r \frac{n}{n\sum x^2 - (\sum y)^2} \tag{B.17}$$

Le terme non diagonal, V_{01} (égal à V_{10} puisque la matrice est symétrique), représente la covariance des deux paramètres. Leur coefficient de corrélation r_{01} est :

$$Cov(b_0, b_1) = V_{01} = V_{10} = V_r \frac{-\sum x}{n\sum x^2 - \left(\sum x\right)^2} \tag{B.18}$$

$$r_{01} = \frac{V_{01}}{\sqrt{V_{00}V_{11}}}$$

B.2 La régression linéaire multiple

Il est supposé ici que la variable dépendante y est en fonction de m variables $x_1, \ldots x_m$:

$$\hat{y} = b_0 + b_1 x_{1k} + b_2 x_{2k} + \cdots + b_m x_{mk} \tag{B.19}$$

Les x_i peuvent être des variables indépendantes ou des fonctions d'une même variable x, par exemple dans la régression polynomiale (où $x_i = x^i$) :

$$\hat{y} = b_0 + b_1 x_k + b_2 x_k^2 + \cdots + b_m x_k^m \tag{B.20}$$

Les paramètres b_i sont estimés en minimisant la somme des carrés des écarts résiduels [Wasserman 2004] :

$$SS_r = \sum_{k=1}^{n} (y_k - b_0 - b_1 x_{1k} - b_2 x_{2k} - \cdots - b_m x_{mk})^2 \tag{B.21}$$

Où n désigne le nombre d'observations et x_i la valeur prise par la variable x_i pour l'observation k. Les formules de la régression linéaire simple sont encore valables sous leur forme matricielle

$$B = A^{-1}C = (U^T U)^{-1}(U^T Y) \tag{B.22}$$

Avec

$$U = X^T$$

$$X = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ x_{11} & x_{12} & x_{13} & \ldots & x_{1n} \\ x_{11} & x_{12} & x_{13} & \ldots & x_{1n} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{m1} & x_{m2} & x_{m3} & \ldots & x_{mn} \end{pmatrix}$$

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \ldots y_n \end{pmatrix}$$

Et :

$$A = \begin{pmatrix} n & \sum x_1 & \dots & \sum x_m \\ \sum x_1 & \sum x_1^2 & \dots & \sum x_1 x_m \\ \dots & \dots & \dots & \dots \\ \sum x_1 & \sum x_m x_1 & \dots & \sum x_m^2 \end{pmatrix} \tag{B.23}$$

$$C = \begin{pmatrix} \sum y_1 \\ \sum x_1 y \\ \dots \\ \sum x_m y \end{pmatrix} \tag{B.24}$$

L'analyse de variance s'effectue comme dans le cas de la régression linéaire simple, sauf qu'ici $p = m + 1$ (nombre de paramètres estimé). Les variances des paramètres sont données par les termes diagonaux de la matrice de variance-covariance, $V = V_r A^{-1}$, les autres termes donnant les covariances. Les coefficients de corrélation entre paramètres sont :

$$r_{jj} = \frac{V_{ij}}{\sqrt{V_{ii} V_{jj}}} \tag{B.25}$$

Les valeurs de r^2, s_r et F se calculent comme précédemment. Toutefois, le coefficient de corrélation r est ici toujours positif. Le coefficient de détermination ajusté est défini par : $r_a^2 = 1 - (1 - r^2)\Delta\frac{n-1}{n-p}$. Lorsque le nombre de paramètres (p) tend vers le nombre d'observations (n), ce coefficient tend vers 0, alors que r^2 tend vers 1. L'examen de la valeur de r_a^2 permet d'éviter l'utilisation d'un trop grand nombre de variables explicatives.

B.3 La régression non linéaire

Une variante de la régression multiple peut quelquefois être appliquée pour ajuster une variable explicative x (ou plusieurs variables explicatives x_j) à une variable dépendante y de manière non linéaire :

$$\hat{y_k} = f(x_k, B) \tag{B.26}$$

Soit par exemple pour une fonction exponentielle :

$$\hat{y_k} = b_0 \,.\, exp(-b_1 \,.\, x_k) \tag{B.27}$$

Cette méthode consiste à ajouter à la variable explicative x de nouvelles variables construites en mettant x au carré, au cube, ou à appliquer des transformations non linéaire aux variables [Seber 2003].

Pour la régression polynomiale, le modèle s'écrit sous forme de polynôme de degré k reliant la (ou les) variable(s) explicative(s) à la variable expliquée ; l'ajout d'un ordre supplémentaire permet d'ajouter un pli à la courbe et d'avoir donc éventuellement un meilleur ajustement graphique, qui se confirmerait par un coefficient de détermination élevé.

$$\hat{y} = b_0 + b_1 x_k + b_2 x_k^2 + \cdots + b_m x_k^m \tag{B.28}$$

Il est important de comprendre ici, que sur le plan technique, les calculs d'une régression non linéaire sont exactement les mêmes que ceux d'une régression multiple traditionnelle. Soit le modèle de régression simple sous la forme :

$$y_k = \hat{y_k} + e_k \tag{B.29}$$

Où e_k représente l'erreur de mesure de y_k (encore appelée "résidu"). Les calculs précédents supposent que :

1. x est connu sans erreur.
2. L'erreur de mesure e_k est constante et indépendante de y.

L'interprétation probabiliste consiste à considérer l'erreur e_k comme une variable aléatoire distribuée selon une loi normale de moyenne 0 et d'écart-type s. Un modèle de régression est donc accepté, une fois que les conditions sur les résidus sont vérifiées [Seber 2003].

Bibliographie

[Abdullah 2006] Ahsan Abdullah et Amir Hussain. *A new biclustering technique based on crossing minimization.* Neurocomputing, vol. 69, no. 16-18, pages 1882–1896, 2006. (Cité en page 37.)

[Achtert 2011] Elke Achtert, Ahmed Hettab, Hans-Peter Kriegel, Erich Schubert et Arthur Zimek. *Spatial outlier detection : data, algorithms, visualizations.* In Proceedings of the 12th international conference on Advances in spatial and temporal databases, SSTD'11, pages 512–516, Berlin, Heidelberg, 2011. Springer-Verlag. (Cité en page 63.)

[Agarwal 2006] Deepak Agarwal, Andrew McGregor, Jeff M. Phillips, Suresh Venkatasubramanian et Zhengyuan Zhu. *Spatial scan statistics : approximations and performance study.* In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '06, pages 24–33, New York, NY, USA, 2006. ACM. (Cité en page 57.)

[Aggarwal 2001] Charu C. Aggarwal et Philip S. Yu. *Outlier detection for high dimensional data.* In Proceedings of the 2001 ACM SIGMOD international conference on Management of data, SIGMOD '01, pages 37–46, New York, NY, USA, 2001. ACM. (Cité en page 61.)

[Andrew Moore 2003] Weng-Keen Wong Andrew Moore. *Optimal Reinsertion : A new search operator for accelerated and more accurate Bayesian network structure learning.* In T. Fawcett et N. Mishra, editeurs, Proceedings of the 20th International Conference on Machine Learning (ICML '03), pages 552–559, Menlo Park, California, August 2003. AAAI Press. (Cité en page 56.)

[Angiulli 2006] Fabrizio Angiulli, Eugenio Cesario et Clara Pizzuti. *A Greedy Search Approach to Co-clustering Sparse Binary Matrices.* In ICTAI, pages 363–370. IEEE Computer Society, 2006. (Cité en page 37.)

[Ankerst 1999] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel et Jörg Sander. *OPTICS : ordering points to identify the clustering structure.* In Proceedings of the 1999 ACM SIGMOD international conference on Management of data, SIGMOD '99, pages 49–60, New York, NY, USA, 1999. ACM. (Cité en page 24.)

[Ayadi 2012] Wassim Ayadi, Mourad Elloumi et Jin-Kao Hao. *Pattern-driven neighborhood search for biclustering of microarray data.* BMC Bioinformatics, vol. 13, no. S-7, page S11, 2012. (Cité en page 37.)

[Babacan 2012] S. DERIN Babacan, Shinichi Nakajima et Minh Do. *Probabilistic Low-Rank Subspace Clustering*. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou et K.Q. Weinberger, editeurs, Advances in Neural Information Processing Systems 25, pages 2753–2761. 2012. (Cité en page 25.)

[Baligh 2012] M. Baligh. *Analyse de données multivariées et surveillance des processus industriels par analyse en composantes principales*. PhD thesis, Université de Saint-Jérôme, France, 2012. (Cité en page 69.)

[Barnett 1978] V. Barnett et T. Lewis. Outliers in statistical data. John Wiley & Sons Ltd., 2nd edition édition, 1978. (Cité en page 65.)

[Basu 2008] Sugato Basu, Ian Davidson et Kiri Wagstaff. Constrained clustering : Advances in algorithms, theory, and applications. Chapman & Hall/CRC, 1 édition, 2008. (Cité en page 15.)

[Ben-Dor 2002] Amir Ben-Dor, Benny Chor, Richard M. Karp et Zohar Yakhini. *Discovering local structure in gene expression data : the order-preserving submatrix problem*. In RECOMB, pages 49–57, 2002. (Cité en pages 49 et 50.)

[Benabdeslem 2012] K. Benabdeslem et K. Allab. *Bi-clustering continuous data with self-organizing map*. Neural Computing and Applications, 2012. (Cité en page 43.)

[Berkhin 2006] P. Berkhin. *A Survey of Clustering Data Mining Techniques*. Grouping Multidimensional Data, pages 25–71, 2006. (Cité en page 14.)

[Bhattacharyya 2011] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel et J. Christopher Westland. *Data mining for credit card fraud : A comparative study*. Decis. Support Syst., vol. 50, no. 3, pages 602–613, Février 2011. (Cité en page 56.)

[Bishop 1994] Chris M. Bishop. *Novelty Detection and Neural Network Validation*, 1994. (Cité en page 56.)

[Borisyuk 2004a] Roman Borisyuk et Yakov B. Kazanovich. *Oscillatory model of attention-guided object selection and novelty detection*. Neural Networks, vol. 17, no. 7, pages 899–915, 2004. (Cité en page 63.)

[Borisyuk 2004b] Roman M. Borisyuk et Yakov B. Kazanovich. *Oscillatory model of attention-guided object selection and novelty detection*. Neural Netw., vol. 17, no. 7, pages 899–915, 2004. (Cité en page 74.)

[Boubou 2003] Mounzer Boubou. Mounzer boubou. Dordrecht, France, 2003. (Cité en page 69.)

[Bouchachia 2011] Abdelhamid Bouchachia. *Fuzzy classification in dynamic environments*. Soft Comput., vol. 15, no. 5, pages 1009–1022, 2011. (Cité en page 35.)

[Bouchachia 2013] Abdelhamid Bouchachia, Edwin Lughofer et Daniel Sanchez. *Editorial of the special issue : Online fuzzy machine learning and data mining*. Inf. Sci., vol. 220, pages 1–4, 2013. (Cité en page 35.)

[Bouchon-Meunier 2010] Bernadette Bouchon-Meunier, Giulianella Coletti, Marie-Jeanne Lesot et Maria Rifqi. *Towards a Conscious Choice of a Fuzzy Similarity Measure : A Qualitative Point of View*. In IPMU, pages 1–10, 2010. (Cité en page 15.)

[Boudjeloud 2005] L. Boudjeloud et F. Poulet. *Visual Interactive Evolutionary Algorithm for High Dimensional Data Clustering and Outlier Detection*. LNCS-3518, Springer-Verlag, pages 426–431, 2005. (Cité en page 66.)

[Box 1965] G. Box et C. Tiao. A change in level of a nonstationary t.s, volume 52. Biometrika, 1965. (Cité en page 61.)

[Box 1990] George Edward Pelham Box et Gwilym Jenkins. Time series analysis, forecasting and control. Holden-Day, Incorporated, 1990. (Cité en page 62.)

[Boyd 2004] Stephen Boyd et Lieven Vandenberghe. Convex optimization. Cambridge University Press, New York, NY, USA, 2004. (Cité en page 52.)

[Breunig 2000] M. Breunig, H. Kriege, R. Ng et J. Sander. *LOF : Identifying Density-Based Local Outliers*. ACM SIGMOD 2000 International congerence on Management of Data, 2000. (Cité en page 60.)

[Busygin 2002] Stanislav Busygin, Gerrit Jacobsen, Ewald Kremer et Contentsoft Ag. *Double Conjugated Clustering Applied to Leukemia Microarray Data*. In In 2nd SIAM ICDM, Workshop on clustering high dimensional data, 2002. (Cité en page 43.)

[Caldas 2011] José Caldas et Samuel Kaski. *Hierarchical Generative Biclustering for MicroRNA Expression Analysis*. Journal of Computational Biology, vol. 18, no. 3, pages 251–261, 2011. (Cité en page 47.)

[Cameron 1997] A. Colin Cameron et Frank Windmeijer. *An R-squared measure of goodness of fit for some common nonlinear regression models*. Journal of Econometrics, vol. 77, no. 2, pages 329–342, 1997. (Cité en page 149.)

[Campbell 1980] N. A. Campbell. *Robust procedures in multivariate analysis I : robust covariance estimation*. Applied Statistics, vol. 29, no. 3, pages 231–237, 1980. (Cité en page 68.)

[Candès 2011] Emmanuel J. Candès, Xiaodong Li, Yi Ma et John Wright. *Robust principal component analysis ?* J. ACM, vol. 58, no. 3, pages 11 :1–11 :37, 2011. (Cité en page 68.)

[Cao 2012] Feng Cao et Soumya Ray. *Bayesian Hierarchical Reinforcement Learning.* In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou et K.Q. Weinberger, editeurs, Advances in Neural Information Processing Systems 25, pages 73–81. 2012. (Cité en page 14.)

[Cattell 1966] R. Cattell. *The scree test for the number of factors.* M.B.R, vol. 1, pages 245–276, 1966. (Cité en pages 85 et 86.)

[Chandola 2009] Varun Chandola, Arindam Banerjee et Vipin Kumar. *Anomaly detection : A survey.* ACM Comput. Surv., vol. 41, no. 3, pages 15 :1–15 :58, Juillet 2009. (Cité en page 74.)

[Charrad 2008] Lechevallier Y. Saporta G. Ben Ahmed-M Charrad M. *Le bipartitionnement : Etat de l'art sur les approches et les algorithmes.* In Ecol'IA 2008, 2008. (Cité en pages 37, 47, 49 et 50.)

[Chen 2005] Da Chen, Xueguang Shao, Bin Hu et Qingde Su. *Simultaneous wavelength selection and outlier detection in multivariate regression of near-infrared spectra.* Analytical sciences the international journal of the Japan Society for Analytical Chemistry, vol. 21, no. 2, pages 161–166, 2005. (Cité en page 56.)

[Cheng 2000] Yizong Cheng et George M. Church. *Biclustering of Expression Data*, 2000. (Cité en pages 49 et 50.)

[Cleuziou 2004] Guillaume Cleuziou, Lionel Martin et Christel Vrain. *PoBOC : An Overlapping Clustering Algorithm, Application to Rule-Based Classification and Textual Data.* In ECAI, pages 440–444, 2004. (Cité en page 15.)

[Cleuziou 2008] Guillaume Cleuziou. *An extended version of the k-means method for overlapping clustering.* In 19th International Conference on Pattern Recognition (ICPR 2008), December 8-11, 2008, Tampa, Florida, USA, pages 1–4. IEEE, 2008. (Cité en pages 15 et 16.)

[Cleuziou 2013] Guillaume Cleuziou. *Osom : A method for building overlapping topological maps.* Pattern Recogn. Lett., vol. 34, no. 3, pages 239–246, Février 2013. (Cité en page 15.)

[Cormack 1971] R. Cormack. *A review of classification.* Journal of the Royal Statistical Society. Series A (General), vol. 134, no. 3, pages 321–367, 1971. (Cité en pages 14 et 15.)

[Cornillon 2007] Pierre-André Cornillon et Eric Matzner-Lø ber. Regression. theory and application. (régression. théorie et applications.). Paris : Springer, 2007. (Cité en page 148.)

[Cottrell 2004] Marie Cottrell, Smail Ibbou et Patrick Letrémy. *SOM-based algorithms for qualitative variables*. Neural Netw., vol. 17, no. 8-9, pages 1149–1167, oct 2004. (Cité en pages 43, 44 et 45.)

[Cox 2006] D.R. Cox. Principles of statistical inference. Cambridge University Press, 2006. (Cité en page 149.)

[De França 2013] F. O. De França, G. P. Coelho et F. J. Von Zuben. *Predicting missing values with biclustering : A coherence-based approach*. Pattern Recogn., vol. 46, no. 5, pages 1255–1266, Mai 2013. (Cité en page 37.)

[Dempster 1977] A. P. Dempster, N. M. Laird et D. B. Rubin. *Maximum likelihood from incomplete data via the EM algorithm*. JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B, vol. 39, no. 1, pages 1–38, 1977. (Cité en pages 26, 27 et 42.)

[Diday 1980] E Diday, G. Govaert, Y. Lechevallier et J. Sidi. *Clustering in Pattern Recognition*. In NATO Conference Series, (Series) 4 : Marine Sciences, pages 424–429, Holland, July 1980. Dordrecht. (Cité en page 16.)

[E. Schubert 2012] H.-P. Kriegel E. Schubert A. Zimek. *Local Outlier Detection Reconsidered : a Generalized View on Locality with Applications to Spatial, Video, and Network Outlier Detection*. Data Mining and Knowledge Discovery, 2012. (Cité en page 61.)

[Eisen 1998] M.B. Eisen, P.T. Spellman, P.O. Brown et D. Botstein. *Cluster analysis and display of genome-wide expression patterns*, 1998. (Cité en page 107.)

[Ester 1996] Martin Ester, Hans-Peter Kriegel, Joerg Sander et Xiaowei Xu. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. In Evangelos Simoudis, Jiawei Han et Usama M. Fayyad, editeurs, Second International Conference on Knowledge Discovery and Data Mining, pages 226–231. AAAI Press, 1996. (Cité en page 24.)

[Evgeniou 2004] Theodoros Evgeniou, Massimiliano Pontil et André Elisseeff. *Leave One Out Error, Stability, and Generalization of Voting Combinations of Classifiers*. Machine Learning, vol. 55, no. 1, pages 71–97, 2004. (Cité en page 122.)

[Fabrizio 2002] A. Fabrizio et C. Pizzuti. *Fast Outlier Detection in High Dimensional Spaces*. PKDD '02 Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, 2002. (Cité en page 59.)

[Favoreel 1998] Wouter Favoreel, Bart De Moor et Peter Van Overschee. *Subspace State Space System Identification For Industrial Processes*, 1998. (Cité en page 63.)

[Fayyad 1996] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth et Ramasamy Uthurusamy, editeurs. Advances in knowledge discovery and data mining. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996. (Cité en page 10.)

[Fiori 2012] Marcelo Fiori, Pablo Mus et Guillermo Sapiro. *Topology Constraints in Graphical Models*. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou et K.Q. Weinberger, editeurs, Advances in Neural Information Processing Systems 25, pages 800–808. 2012. (Cité en page 15.)

[Forgy 1965] E. Forgy. *Cluster analysis of multivariate data : efficiency versus interpretability of classifications*. Biometrics, vol. 21, pages 768–780, 1965. (Cité en page 16.)

[Frank 2010] A. Frank et A. Asuncion. *UCI machine learning repository*. Technical report, School of Information and Computer Sciences, available at :http ://archive.ics.uci.edu/ml, 2010. (Cité en pages 81 et 106.)

[Freitas 1985] R.K. Freitas. K-théorie réelle des variétés de stiefel sans torsion. 1985. (Cité en page 53.)

[Gao 2010] J. Gao, W. Hu, W. Li et Z. Zhang. *Local Outlier Detection Based on Kernel Regression*. International Conference on Pattern Recognition, 2010. (Cité en page 59.)

[Getz 2000a] G. Getz, E. Levine et E. Domany. *Coupled Two-Way Clustering Analysis of Gene Microarray Data*. Proc. Natl. Acad. Sci. USA, vol. 97, pages 12079–12084, 2000. (Cité en pages 47 et 109.)

[Getz 2000b] G. Getz, E. Levine, E. Domany et M. Q. Zhang. *Super Paramagnetic Clustering of Yeast Gene Expression Profiles*, 2000. (Cité en page 47.)

[Gordon 2012] N.J. Gordon, D.J. Salmond et A.F.M. Smith. *Novel approach to nonlinear/non-Gaussian Bayesian state estimation*. IEEE Proceedings F, Radar and Signal Processing, vol. 140, no. 2, pages 107–113, 2012. (Cité en page 63.)

[Govaert 1983] G. Govaert. *Classification croisée*. PhD thesis, Université Paris 6, France, 1983. (Cité en pages 16, 37, 38, 100 et 105.)

[Govaert 2009] Gerard Govaert et Mohamed Nadif. *Un modèle de mélange pour la classification croisée d'un tableau de données continues*. 2009. (Cité en pages 37, 40 et 41.)

[Greene 2010] D. Greene et P. Cunningham. *Spectral co-clustering for dynamic bipartite graphs*. In Workshop on dynamic networks and knowledge discovery at ecml'10, barcelona, spain, 2010. (Cité en page 37.)

[Guha 1998] Sudipto Guha, Rajeev Rastogi et Kyuseok Shim. *CURE : an efficient clustering algorithm for large databases*. In Proceedings of the 1998 ACM SIGMOD international conference on Management of data, SIGMOD '98, pages 73–84, New York, NY, USA, 1998. ACM. (Cité en page 20.)

[Guyon 2003] Isabelle Guyon et André Elisseeff. *An introduction to variable and feature selection*. J. Mach. Learn. Res., vol. 3, pages 1157–1182, Mars 2003. (Cité en page 122.)

[Hamel 2009] L. Hamel. Knowledge discovery with support vector machines. Wiley-Interscience, 2009. (Cité en pages 71 et 72.)

[Hartigan 1972] J. A. Hartigan. *Direct Clustering of a Data Matrix*. Journal of the American Statistical Association, vol. 67, no. 337, pages 123–129, 1972. (Cité en pages 37 et 46.)

[Hartigan 1979] J. A. Hartigan et M. A. Wong. *A k-means clustering algorithm*. JSTOR : Applied Statistics, vol. 28, no. 1, pages 100–108, 1979. (Cité en page 15.)

[Hasan 2009] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem et Mohammed J. Zaki. *Robust partitional clustering by outlier and density insensitive seeding*. Pattern Recogn. Lett., vol. 30, pages 994–1002, August 2009. (Cité en page 60.)

[Hastie 2009] T.J. Hastie, R.J. Tibshirani et J.J.H. Friedman. The elements of statistical learning : Data mining, inference, and prediction. Springer series in statistics. Springer-Verlag New York, 2009. (Cité en page 10.)

[Hawkins 1980] D. M. Hawkins. Identification of outliers. Monographs on applied probability and statistics. Chapman and Hall, London [u.a.], 1980. (Cité en page 65.)

[Hoffmann 2007] Heiko Hoffmann. *Kernel PCA for novelty detection*. Pattern Recognition, vol. 40, 2007. (Cité en pages 70 et 88.)

[Huber 2009] P.J. Huber et E.M. Ronchetti. Robust statistics. Wiley Series in Probability and Statistics. Wiley, 2009. (Cité en page 68.)

[Hubert 2003] Mia Hubert, Peter J. Rousseeuw et Karlien Vanden Branden. *ROBPCA : a New Approach to Robust Principal Component Analysis*, 2003. (Cité en page 71.)

[Jagota 1991] A. Jagota. *Novelty detection on a very large number of memories stored in a Hopfield-style network*. In Proceedingsof the Interna-

tional Joint Conference on Neural Networks, volume 2, Seattle, WA, 1991. (Cité en page 74.)

[Jahirabadkar 2013] Sunita Jahirabadkar et Parag Kulkarni. *Article : Clustering for High Dimensional Data : Density based Subspace Clustering Algorithms*. International Journal of Computer Applications, vol. 63, no. 20, pages 29–35, February 2013. Published by Foundation of Computer Science, New York, USA. (Cité en page 24.)

[Jain 1999] A. K. Jain, M. N. Murty et P. J. Flynn. *Data clustering : a review*. ACM Comput. Surv., vol. 31, no. 3, pages 264–323, Septembre 1999. (Cité en pages 13 et 22.)

[Jolliffe 1986] I.T. Jolliffe. Principal component analysis. Springer Verlag, 1986. (Cité en page 70.)

[Jollois 2003] Xavier Jollois. Contribution de la classification automatique à la fouille de données. Dordrecht, France, 2003. (Cité en pages 26, 39 et 46.)

[Kalman 1960] R. E. Kalman. *A New Approach to Linear Filtering and Prediction Problems*. 1960. (Cité en page 62.)

[Kaur 2013] Prabhjot Kaur, A. K. Soni et Anjana Gosain. *A robust kernelized intuitionistic fuzzy c-means clustering algorithm in segmentation of noisy medical images*. Pattern Recognition Letters, vol. 34, no. 2, pages 163–175, 2013. (Cité en page 15.)

[Kaustav Das 2008] Jeff Schneider Kaustav Das et Daniel Neill. *Anomaly Pattern Detection in Categorical Datasets*. In Proceedings of 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008), August 2008. (Cité en page 57.)

[Knorr 1997] Edwin M. Knorr et Raymond T. Ng. *A Unified Notion of Outliers : Properties and Computation*. In In Proc. of the International Conference on Knowledge Discovery and Data Mining, pages 219–222. AAAI Press, 1997. (Cité en page 59.)

[Knorr 1998] Edwin M. Knorr et Raymond T. Ng. *Algorithms for Mining Distance-Based Outliers in Large Datasets*. In VLDB, pages 392–403, 1998. (Cité en page 66.)

[Kohavi 1998] Kohavi et Provost. *Glossary of terms*. Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process, 1998. (Cité en pages 81 et 88.)

[Kohonen 1995] Kohonen. Self-organizing maps. Springer Verlag, Berlin, 1995. (Cité en pages 15 et 28.)

[Kohonen 2001] T. Kohonen, M. R. Schroeder et T. S. Huang, editeurs. Self-organizing maps. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd édition, 2001. (Cité en pages 43 et 107.)

[Kriegel 2011] Hans-Peter Kriegel, Peer Kröger, Jörg Sander et Arthur Zimek. *Density-based clustering.* Wiley Interdisc. Rew. : Data Mining and Knowledge Discovery, vol. 1, no. 3, pages 231–240, 2011. (Cité en page 24.)

[Kundu 2012] Malay K. Kundu, Sushmita Mitra, Debasis Mazumdar et Sankar K. Pal, editeurs. Perception and machine intelligence - first indo-japan conference, permin 2012, kolkata, india, january 12-13, 2012. proceedings, volume 7143 of *Lecture Notes in Computer Science*. Springer, 2012. (Cité en page 15.)

[Kwon 2010] Bongjune Kwon et Hyuk Cho. *Scalable Co-clustering Algorithms.* In ICA3PP (1), pages 32–43, 2010. (Cité en page 37.)

[Labiod 2011] Lazhar Labiod et Mohamed Nadif. *Co-clustering under nonnegative matrix tri-factorization.* In Proceedings of the 18th international conference on Neural Information Processing - Volume Part II, ICONIP'11, pages 709–717, Berlin, Heidelberg, 2011. Springer-Verlag. (Cité en pages 50, 51 et 109.)

[Lazzeroni 2000] Laura Lazzeroni et Art Owen. *Plaid Models for Gene Expression Data.* Statistica Sinica, vol. 12, pages 61–86, 2000. (Cité en pages 49 et 50.)

[Lee 1999] D. D. Lee et H. S. Seung. *Learning the Parts of Objects by Nonnegative Matrix Factorization.* Nature, vol. 401, page 788, 1999. (Cité en pages 34, 35 et 51.)

[Lee 2013] Yi-Ren ; Wang Yu-Chiang Frank Lee Yuh-Jye ; Yeh. *Anomaly detection via online oversampling principal component analysis.* volume 25, pages 1460–1470, 2013. (Cité en page 70.)

[Levillain 2010] Florent Levillain, Joseph Onderi Orero, Maria Rifqi et Bernadette Bouchon-Meunier. *Characterizing player's experience from physiological signals using fuzzy decision trees.* In CIG, pages 75–82, 2010. (Cité en page 15.)

[ling Shyu 2003] Mei ling Shyu, Shu ching Chen, Kanoksri Sarinnapakorn et Liwu Chang. *A novel anomaly detection scheme based on principal component classifier.* In in Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03, pages 172–179, 2003. (Cité en pages 70 et 71.)

[Liu 1998] Jun S. Liu et Rong Chen. *Sequential Monte Carlo Methods for Dynamic Systems.* Journal of the American Statistical Association, vol. 93, pages 1032–1044, 1998. (Cité en page 63.)

[Long 2005] Bo Long, Zhongfei (Mark) Zhang et Philip S. Yu. *Co-clustering by block value decomposition.* In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, KDD '05, pages 635–640, New York, NY, USA, 2005. ACM. (Cité en pages 50, 53 et 109.)

[Mahmoud 2012] Sawsan M. Mahmoud, Ahmad Lotfi et Caroline Langensiepen. *User activities outlier detection system using principal component analysis and fuzzy rule-based system.* In Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments, PETRA '12, pages 26 :1–26 :8, New York, NY, USA, 2012. ACM. (Cité en page 70.)

[Manski 1989] C.F. Manski et University of Wisconsin-Madison. Social Systems Research Institute. Regression. SSRI workshop series. Social Systems Research Institute, University of Wisconsin, 1989. (Cité en page 147.)

[Mao 2005] D. Mao, Y. Luo, J. Zhang et J. Zhu. *A new strategy of cooperativity of biclustering and hierarchical clustering : a case of analyzing yeast genomic microarray datasets.* Front Biosci, vol. 10, 2005. (Cité en page 47.)

[Marascu 2009] Alice Marascu et Florent Masseglia. *Détection d'enregistrements atypiques dans un flot de données : une approche multi-résolution.* In EGC, pages 455–456, 2009. (Cité en page 66.)

[Markou 2003a] Markos Markou et Sameer Singh. *Novelty detection : a review part 1 : statistical approaches.* Signal Process., vol. 83, pages 2481–2497, December 2003. (Cité en pages 67 et 68.)

[Markou 2003b] Markos Markou et Sameer Singh. *Novelty detection : a review part 2 : neural network based approaches.* Signal Process., vol. 83, pages 2499–2521, December 2003. (Cité en pages 74 et 88.)

[Marlin 2012] Benjamin M. Marlin, David C. Kale, Robinder G. Khemani et Randall C. Wetzel. *Unsupervised pattern discovery in electronic health care data using probabilistic clustering models.* In Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, IHI '12, pages 389–398, New York, NY, USA, 2012. ACM. (Cité en page 25.)

[McCullagh 2009] Peter McCullagh, Vladimir Vovk, Ilia Nouretdinov, Dmitry Devetyarov et Alexander Gammerman. *Conditional Prediction Inter-*

vals for Linear Regression. In ICMLA, pages 131–138, 2009. (Cité en page 124.)

[Mennatallah Amer 2012] Markus Goldstein Mennatallah Amer. *Nearest-Neighbor and Clustering based Anomaly Detection Algorithms for RapidMiner*. pages 1–12. Shaker Verlag GmbH, Aachen, 8 2012. (Cité en page 61.)

[Meyer 2011] G. Meyer. *Geometric optimization algorithms for linear regression on fixed-rank matrices*. PhD thesis, University of Liège, 2011. (Cité en page 149.)

[Montgomery 2007] D.C. Montgomery. Introduction to statistical quality control, 4th ed. Wiley India Pvt. Limited, 2007. (Cité en page 62.)

[Moya 1993] M. R. Moya, M. W. Koch et L. D. Hostetler. *One-class classifier networks for target recognition applications*. World Congress on Neural Networks, International Neural Network Society (INNS), 1993. (Cité en page 74.)

[Munaga 2012] Hazarath Munaga, M. D. R. Mounica Sree et J. V. R. Murthy. *Article : DenTrac : A Density based Trajectory Clustering Tool*. International Journal of Computer Applications, vol. 41, no. 10, pages 17–21, March 2012. Published by Foundation of Computer Science, New York, USA. (Cité en page 24.)

[Murtagh 1983] F. Murtagh. *A survey of recent advances in hierarchical clustering algorithms*. Computer Journal, vol. 26, no. 4, pages 354–359, 1983. (Cité en page 22.)

[Nadif 2004] M. Nadif, F.-X. Jollois et G. Govaert. *Block Clustering for large continuous data sets*. In AISTA'2004, International Conference on Advances in Intelligent Systems - Theory and Applications in cooperation with IEEE Computer Society, (in CD ISBN 2-9599776-8-8), Luxembourg, 15-18 November 2004. (Cité en page 40.)

[Nagpal 2011] Pooja Batra Nagpal et Priyanka Ahlawat Mann. *Article : Comparative Study of Density based Clustering Algorithms*. International Journal of Computer Applications, vol. 27, no. 11, pages 44–47, August 2011. Published by Foundation of Computer Science, New York, USA. (Cité en page 24.)

[Odin 2000] T. Odin et Addison D. *Novelty detection using neural network technology*. 2000. (Cité en page 68.)

[Paatero 1994] P. Paatero et U. Tapper. *Positive Matrix Factorization : A Non-Negative Factor Model with Optimal Utilization of Error Estimates of Data Values*. pages 111–126, 1994. (Cité en page 107.)

[Pal 2005] N. R. Pal, K. Pal, J. M. Keller et J. C. Bezdek. *A Possibilistic Fuzzy c-Means Clustering Algorithm.* Trans. Fuz Sys., vol. 13, no. 4, pages 517–530, Août 2005. (Cité en page 16.)

[Papadimitriou 2003] Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B. Gibbons et Christos Faloutsos. *LOCI : Fast Outlier Detection Using the Local Correlation Integral.* In ICDE, pages 315–326, 2003. (Cité en page 62.)

[Patel 2011] Pranav Patel, Eamonn Keogh, Jessica Lin et Stefano Lonardi. *Mining Motifs in Massive Time Series Databases.* In Proceedings of the 2011 IEEE International Conference on Data Mining, ICDM '11, pages 370–, Washington, DC, USA, 2011. IEEE Computer Society. (Cité en page 63.)

[Pearson 1901] K. Pearson. On lines and planes of closest fit to systems of points in space. University College, 1901. (Cité en page 69.)

[Pekalska 2003] Elzbieta Pekalska, David M. J. Tax et Robert P. W. Duin. *One-class LP classifiers for dissimilarity representations.* In ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, pages 761–768. MIT Press, 2003. (Cité en page 87.)

[Planchon 2007] Viviane Planchon. Détection de valeurs aberrantes dans des mélanges de distributions dissymétriques pour des ensembles de données avec contraintes spatiales. 2007. (Cité en page 58.)

[Plantevit 2007] Marc Plantevit, Anne Laurent et Maguelonne Teisseire. *Extraction d'outliers dans des cubes de données : une aide à la navigation.* In Ladjel Bellatreche, Arnaud Giacometti et Patrick Marcel, editeurs, EDA, volume B-3 of *RNTI*, pages 113–130. Cépadus, 2007. (Cité en page 59.)

[Powers 2007] David M. W. Powers. Rapport technique SIE-07-001, School of Informatics and Engineering, Flinders University, Adelaide, Australia, 2007. (Cité en page 82.)

[Ramaswamy 2000] Sridhar Ramaswamy, Rajeev Rastogi et Kyuseok Shim. *Efficient algorithms for mining outliers from large data sets.* SIGMOD Rec., vol. 29, no. 2, pages 427–438, 2000. (Cité en page 59.)

[Ringberg 2007] Haakon Ringberg, Augustin Soule, Jennifer Rexford et Christophe Diot. *Sensitivity of PCA for traffic anomaly detection.* SIGMETRICS Perform. Eval. Rev., vol. 35, no. 1, pages 109–120, Juin 2007. (Cité en page 70.)

[Rusiecki 2012] Andrzej Rusiecki. *Robust Neural Network for Novelty Detection on Data Streams.* In ICAISC (1), pages 178–186, 2012. (Cité en page 74.)

[Saporta 1990] G. Saporta. Probabilités analyse des données et statistique, chapitre No 5 Notions élémentaires sur les processus aléatoires, pages 103–113. Technip, 1990. (Cité en page 70.)

[Saporta 2006] G. Saporta. Probabilités, analyses des données et statistiques. Editions Technip, 2006. (Cité en pages 10 et 13.)

[Sarawagi 1998] Sunita Sarawagi, Rakesh Agrawal et Nimrod Megiddo. *Discovery-driven Exploration of OLAP Data Cubes*. In In Proc. Int. Conf. of Extending Database Technology (EDBT'98, pages 168–182. Springer-Verlag, 1998. (Cité en page 66.)

[Scherrer 2012] Chad Scherrer, Ambuj Tewari, Mahantesh Halappanavar et David Haglin. *Feature Clustering for Accelerating Parallel Coordinate Descent*. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou et K.Q. Weinberger, editeurs, Advances in Neural Information Processing Systems 25, pages 28–36. 2012. (Cité en page 36.)

[Scholkopf 2000] Bernhard Scholkopf, Robert Williamson, Alex Smola, John Shawe-Taylor et John Platt. *Support Vector Method for Novelty Detection*, 2000. (Cité en page 72.)

[Scholkopf 2001] Bernhard Scholkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola et Robert C. Williamson. *Estimating the Support of a High-Dimensional Distribution*. Neural Comput., vol. 13, no. 7, pages 1443–1471, Juillet 2001. (Cité en page 88.)

[Seber 2003] G.A.F. Seber et C.J. Wild. Nonlinear regression. Wiley Series in Probability and Statistics. Wiley, 2003. (Cité en pages 152 et 153.)

[Shan 2010] Hanhuai Shan, et Arindam Banerjee. *Residual Bayesian Co-clustering for Matrix Approximation*. In SDM, pages 223–234, 2010. (Cité en page 37.)

[Shang 2012] Fanhua Shang, L. C. Jiao et Fei Wang. *Graph dual regularization non-negative matrix factorization for co-clustering*. Pattern Recogn., vol. 45, no. 6, pages 2237–2250, Juin 2012. (Cité en page 50.)

[Shekhar 2001] Shashi Shekhar, Chang-Tien Lu et Pusheng Zhang. *Detecting Graph-Based Spatial Outliers : Algorithms and Applications (Summary of Results)*, 2001. (Cité en page 56.)

[Silla 2011] Carlos N. Silla Jr. et Alex A. Freitas. *A survey of hierarchical classification across different application domains*. Data Min. Knowl. Discov., vol. 22, no. 1-2, pages 31–72, Janvier 2011. (Cité en page 18.)

[Singh 2006] Maneesha Singh, Sameer Singh et Markos Markou. *Partial Object Recognition for Improving Novelty Detection in Videos*. In IJCNN, pages 2550–2554, 2006. (Cité en page 74.)

[Song 2007] Xiuyao Song, Mingxi Wu, Christopher Jermaine et Sanjay Ranka. *Conditional Anomaly Detection*. IEEE Trans. on Knowl. and Data Eng., vol. 19, no. 5, pages 631–645, Mai 2007. (Cité en page 57.)

[Strehl 2002] Alexander Strehl, Joydeep Ghosh et Claire Cardie. *Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions*. Journal of Machine Learning Research, vol. 3, pages 583–617, 2002. (Cité en page 105.)

[Sun 2006] Pei Sun, Sanjay Chawla et Bavani Arunasalam. *Mining for outliers in sequential databases*. In in ICDM, 2006, pages 94–106, 2006. (Cité en page 66.)

[Tan 2005] Pang-Ning Tan, Michael Steinbach et Vipin Kumar. Introduction to data mining, (first edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. (Cité en page 10.)

[Tanay 2002] Amos Tanay, Roded Sharan et Ron Shamir. *Discovering Statistically Significant Biclusters in Gene Expression Data*. In In Proceedings of ISMB 2002, pages 136–144, 2002. (Cité en pages 37 et 49.)

[Tong 2002] Christopher Tong et Vladimir Svetnik. *Novelty detection in mass spectral data using a support vector machine method*. In Proc. of Interface, 2002. (Cité en page 71.)

[Tsai 2012] Chen-An Tsai, Chien-Hsun Huang, Ching-Wei Chang et Chun-Houh Chen. *Recursive Feature Selection with Significant Variables of Support Vectors*. Comp. Math. Methods in Medicine, vol. 2012, 2012. (Cité en page 122.)

[Tsay 1988] R.S. Tsay. *Outliers, level shifts, and variance changes in time series*. J.O.F, vol. 7, pages 1–20, 1988. (Cité en page 62.)

[Vapnik 1995] V. N. Vapnik. The nature of statistical learning theory. Springer, New York, 1995. (Cité en pages 14 et 71.)

[Vichi 2001] M. Vichi. *Double k-means Clustering for Simultaneous Classification of Objects and Variables*. pages 43–52, 2001. (Cité en page 16.)

[Vijaya 2006] P. A. Vijaya, M. Narasimha Murty et D. K. Subramanian. *Efficient bottom-up hybrid hierarchical clustering techniques for protein sequence classification*. Pattern Recogn., vol. 39, no. 12, pages 2344–2355, Décembre 2006. (Cité en page 19.)

[Wang 2007] Yin Wang, Jeonghwa Lee et Jun Zhang 0001. *Frobenius norm minimization and probing for preconditioning*. Int. J. Comput. Math., vol. 84, no. 8, pages 1211–1223, 2007. (Cité en page 35.)

[Wasserman 2004] Larr Wasserman. All of statistics : a concise course in statistical inference. 2004. (Cité en page 151.)

[Weng-Keen Wong 2002] Gregory Cooper Michael Wagner Weng-Keen Wong Andrew Moore. *Rule-based Anomaly Pattern Detection for Detecting Disease Outbreaks*. In Proceedings of the 18th National Conference on Artificial Intelligence. MIT Press, 2002. Also available online from http ://www.cs.cmu.edu/*sim*awm/antiterror. (Cité en page 62.)

[Weston 2000] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio et V. Vapnik. *Feature selection for SVMs*. In Advances in Neural Information Processing Systems 13, pages 668–674. MIT Press, 2000. (Cité en page 122.)

[Wichern 1976] D. W. Wichern, R. B. Miller et D. A. Hsu. *Changes of variance in first-order autoregressive time series models with application*. Applied statistics, vol. 25, pages 248–256, 1976. (Cité en page 62.)

[Williams 1959] W.T. Williams et J. Lambert. Multivariate methods in plant ecology. Blackwell Scientific Publications, 1959. (Cité en page 22.)

[Winter 2011] Philipp Winter, Eckehard Hermann et Markus Zeilinger. In NTMS, pages 1–5. IEEE, 2011. (Cité en page 71.)

[Wolfe 1970] John H. Wolfe. *Pattern clustering by multivariate mixture analysis*. Multivariate Behavioral Research, vol. 5, pages 329–350, 1970. (Cité en page 26.)

[Xing 2009] Hong-Jie Xing, Ming-Hu Ha et Xi-Zhao Wang. *Combining SOM and local minimum enclosing spheres for novelty detection*. In Proceedings of the 21st annual international conference on Chinese control and decision conference, CCDC'09, pages 3814–3819, Piscataway, NJ, USA, 2009. IEEE Press. (Cité en page 74.)

[Xu 1998] Xiaowei Xu, Martin Ester, Hans peter Kriegel et Jurg S. *A distribution-based clustering algorithm for mining in large spatial databases*. pages 324–331, 1998. (Cité en page 24.)

[Yoo 2010] Jiho Yoo et Seungjin Choi. *Orthogonal nonnegative matrix tri-factorization for co-clustering : Multiplicative updates on Stiefel manifolds*. Inf. Process. Manage., vol. 46, no. 5, pages 559–570, Septembre 2010. (Cité en page 50.)

[Young 1993] R.K. Young. Wavelet theory and its applications. The Kluwer international series in engineering and computer science. Kluwer Academic Publishers, 1993. (Cité en page 66.)

[Ypma 1997] Alexander Ypma, Er Ypma et Robert P.W. Duin. *Novelty detection using Self-Organizing Maps*. In In Proc. of ICONIP'97, pages 1322–1325. Springer, 1997. (Cité en page 74.)

[Zengyou 2003] He. Zengyou, X. Xu et S. Deng. *Discovering cluster-based local outliers*. Journal Pattern Recognition Letter, vol. 24, pages 9–10, 2003. (Cité en page 61.)

[Zhang 2004] Jian Zhang, Yiming Yang et Jaime Carbonell. *New Event Detection with nearest Neighbor, Support Vector Machines, and Kernel Regression*. Rapport technique CMU-CS-04-118, CMU, April 2004. (Cité en page 71.)

[Zhao 2005] Ying Zhao, George Karypis et Usama Fayyad. *Hierarchical Clustering Algorithms for Document Datasets*. Data Min. Knowl. Discov., vol. 10, no. 2, pages 141–168, 2005. (Cité en page 20.)