Université Claude Bernard Lyon 1
École Doctorale Informatique et Information pour la Société

2007 – 2008

# THÈSE

pour obtenir le grade de

Docteur en Informatique

(arrêté du 7 août 2006)

présentée et soutenue publiquement par

**Haytham ELGHAZEL**

le 17 décembre 2007

## Classification et Prévision des Données Hétérogènes : Application aux Trajectoires et Séjours Hospitaliers

préparée au sein du laboratoire LIESP – Université Claude Bernard Lyon 1

sous la direction de

**M. Alain DUSSAUCHOY**
**Mme Véronique DESLANDRES**

### COMPOSITION DU JURY

| | | |
|---|---|---|
| M. Jean-Pierre BARTHELEMY | Rapporteur | (Professeur, ENST Bretagne) |
| M. Edwin DIDAY | Rapporteur | (Professeur, Université Paris-Dauphine ) |
| M. Younès BENNANI | Examinateur | (Professeur, Université Paris 13) |
| M. Hamamache KHEDDOUCI | Examinateur | (Professeur, Université Lyon 1) |
| M. Michel LAMURE | Examinateur | (Professeur, Université Lyon 1) |
| M. Gilbert SAPORTA | Examinateur | (Professeur, CNAM Paris) |
| M. Alain DUSSAUCHOY | Directeur de thèse | (Professeur, Université Lyon 1) |
| Mme Véronique DESLANDRES | Co-Directrice de thèse | (Maître de conférences, Université Lyon 1) |

# Remerciements

*L'écriture d'une telle page n'est pas une tâche aisée. Déjà trois ans depuis le tout début. Pendant ces trois ans, j'ai rencontré des personnes qui ont contribué à ce projet et à qui j'adresse mes remerciements les plus sincères. Certes des noms viennent immédiatement à l'esprit, mais combien d'anonymes ou d'oubliés y ont, indirectement, contribué. Que ceux et celles que je n'ai pas nommés reçoivent aussi l'expression de toutes mes pensées.*

*Tout d'abord, je tiens à exprimer mes plus vifs remerciements et ma gratitude à mes directeurs de thèse, Mr Alain DUSSAUCHOY et Mme Véronique DESLANDRES, pour leurs encadrements continus, pour les remarques constructives qu'ils m'ont fournies ainsi que pour leurs précieux conseils durant toute la période de mon travail. Je les remercie également pour la confiance qu'ils m'ont accordée et pour la grande liberté d'idées et de travail qu'ils m'ont donnée. En dehors de leurs apports scientifiques, je n'oublierai pas aussi de les remercier pour leurs qualités humaines, leur hospitalité et leur soutien qui m'ont permis de mener à bien cet ouvrage.*

*Je remercie Monsieur Edwin DIDAY, Professeur à l'Université Paris Dauphine, et Monsieur Jean-Pierre BARTHELEMY, Professeur à l'Ecole Nationale Supérieure des Télécommunications de Bretagne, du temps et de l'attention qu'ils ont consacrée pour rapporter cette thèse. Leurs critiques et leurs suggestions m'ont permis d'améliorer mon travail.*

*Ma gratitude, mon profond respect et mes remerciements à tous les membres du jury pour leur travail et l'attention consacrée à l'égard de mon travail.*

*Je remercie les membres des laboratoires LIESP et LIRIS que j'ai pu côtoyer durant la période de ma thèse et qui ont su rendre mon travail agréable, par leur simple présence et l'ambiance qu'ils ont su créer. Je pense à Kaouther, Zahra, Lyes, Mohamed, Amjad et Fazia, pour leur bonne humeur et leur soutien, Khalid pour nos nombreux cafés et le plaisir partagé à discuter de la recherche ensemble. C'était très agréable de travailler en leur compagnie.*

*Ma reconnaissance va aux professeurs Hamamache KHEDDOUCI et Mohand-Saïd HACID, pour leur disponibilité et leurs connaissances dans de nombreux domaines. Ils ont ainsi largement contribué au bon déroulement de mes travaux, et ont ensuite été présents au quotidien pour m'aider à surmonter les diverses difficultés, m'encourager, et me prodiguer de bons conseils.*

*Un grand merci également au professeur Michel LAMURE, à l'origine du financement de cette thèse, en tant que responsable du projet DOMECAD de la région Rhône-Alpes.*

*Je dédie du plus profond de mon cœur ce travail, à mes chers parents Mohamed et Nasra, à mon frère Heni, à ma sœur Hiba. C'est grâce à leur soutien, leur patience et leur amour que je suis là aujourd'hui. Je leur suis très reconnaissant pour les sacrifices qu'ils ont du faire pendant mes longues années d'études et d'absence.*

*Une pensée aux membres de ma grande famille, de ma belle famille, Moufida, Salem et Wissem, à tous ceux que j'aime et qui m'aiment et aux mémoires de mon grand-père Hamda et de ma grand-mère Mbarka.*

*Que mes amis trouvent également ici le témoignage de ma reconnaissance et de mon amitié, pour l'agréable bout de chemin qu'on a passé ensemble, pour leur disponibilité et leur sympathie. Ma reconnaissance va à tous les amis qui ont franchi la méditerranée en même temps que moi et avec qui j'ai passé de bons moments de joie et de détente. Leur présence a compensé en partie l'éloignement de ma famille et de mon pays, tout particulièrement, Néjib MOALLA, Sodki CHÂARI, Tarak CHÂARI, Nizar MNIF, Sabeur ELKOSSANTINI, Mahmoud GHORBEL et Brahim ELLOUMI.*

*Je dédie aussi ce travail à tous ceux qui m'ont apporté leur savoir et contribué à ma formation : mes instituteurs du primaire, mes professeurs du lycée, mes enseignants de l'ENIS. Qu'ils trouvent ici l'expression de ma gratitude et de mon profond respect.*

*Enfin un grand merci à ma femme, Miniar, pour son soutien et le deuxième souffle qu'elle a toujours su apporter avec tendresse et sérénité dans les moments les plus difficiles.*

*Merci à tous ceux qui y ont cru à ce projet, l'ont soutenu et l'ont défendu.*

# Résumé

Ces dernières années ont vu le développement des techniques de fouille de données dans de nombreux domaines d'applications dans le but d'analyser des données volumineuses et complexes. La santé est ainsi un secteur où les données disponibles sont nombreuses et de natures variées (variables classiques comme l'âge ou le sexe, variables symboliques comme l'ensemble des actes médicaux, les diagnostics, etc). D'une manière générale, la fouille de données regroupe l'ensemble des techniques soit descriptives (qui visent à mettre en évidence des informations présentes mais cachées par le volume des données), soit prédictives (cherchant à extrapoler de nouvelles connaissances à partir des informations présentes dans les données). Dans le cadre de cette thèse, nous nous intéressons au problème de classification et de prévision de données hétérogènes, que nous proposons d'étudier à travers deux approches principales. Dans la première, il s'agit de mettre en place une nouvelle approche de classification automatique basée sur une technique de la théorie des graphes baptisée b-coloration. Nous avons également développé l'apprentissage incrémental associé à cette approche, ce qui permet à de nouvelles données d'être automatiquement intégrées dans la partition initialement générée sans avoir à relancer la classification globale. Le deuxième apport de notre travail concerne l'analyse de données séquentielles. Nous proposons de combiner la méthode de classification précédente avec les modèles de mélange markovien, afin d'obtenir une partition de séquences temporelles en groupes homogènes et significatifs. Le modèle obtenu assure une bonne interprétabilité des classes construites et permet d'autre part d'estimer l'évolution des séquences d'une classe donnée.

Ces deux propositions ont ensuite été appliquées aux données issues du système d'information hospitalier français (PMSI), dans l'esprit d'une aide au pilotage stratégique des établissements de soins. Ce travail consiste à proposer dans un premier temps une typologie plus fine des séjours hospitaliers pour remédier aux problèmes associés à la classification existante en groupes homogènes de malades (GHM). Dans un deuxième temps, nous avons cherché à définir une typologie des trajectoires patient (succession de séjours hospitaliers d'un même patient) afin de prévoir de manière statistiques les caractéristiques du prochain séjour d'un patient arrivant dans un établissement de soins. La méthodologie globale offre ainsi un environnement d'aide à la décision pour le suivi et la maîtrise de l'organisation du système des soins.

**Mots clés :** Classification automatique, prévision, classification incrémentale, données hétérogènes, données séquentielles, b-coloration de graphes, séjours hospitaliers, trajectoires hospitalières.

# Abstract

Recent years have seen the development of data mining techniques in various application areas, with the purpose of analyzing large and complex data. The medical field is one of these areas where available data are numerous and described using various attributes, classical (like patient age and sex) or symbolic (like medical treatments and diagnosis). Data mining generally includes either descriptive techniques (which provide an attractive mechanism to automatically find the hidden structure of large data sets), or predictive techniques (able to unearth hidden knowledge from datasets). In this work, the problem of clustering and prediction of heterogeneous data is tackled by a two-stage proposal. The first one concerns a new clustering approach which is based on a graph coloring method, named b-coloring. An extension of this approach which concerns incremental clustering has been added at the same time. It consists in updating clusters as new data are added to the dataset without having to perform complete re-clustering. The second proposal concerns sequential data analysis and provides a new framework for clustering sequential data based on a hybrid model that uses the previous clustering approach and the Mixture Markov chain models. This method allows building a partition of the sequential dataset into cohesive and easily interpretable clusters, as well as it is able to predict the evolution of sequences from one cluster.

Both proposals have then been applied to healthcare data given from the PMSI program (French hospital information system), in order to assist medical professionals in their decision process. In the first step, the b-coloring clustering algorithm has been investigated to provide a new typology of hospital stays as an alternative to the DRGs classification (Diagnosis Related Groups). In a second step, we defined a typology of clinical pathways and are then able to predict possible features of future paths when a new patient arrives at the clinical center. The overall framework provides a decision-aid system for assisting medical professionals in the planning and management of clinical process.

**Keywords:** Clustering, prediction, incremental clustering, heterogeneous data, sequential data, graph b-coloring, hospital stays, clinical pathways.

*“ La recherche procède par des moments distincts et durables, intuition, aveuglement, exaltation et fièvre. Elle aboutit un jour à cette joie, et connaît cette joie celui qui a vécu des moments singuliers "*

Albert Einstein, " *Comment je vois le monde"*

# Table des matières

# Table des figures

# Table des tableaux

# Chapitre 1

# Introduction générale

*" L'analyse des données est un outil pour dégager de la gangue des données le pur diamant de la véridique nature. "*

Jean-Paul Benzécri, "*Histoire et préhistoire de l'analyse des données, 1976*"

## 1.1. Contexte et problématique

Le développement des systèmes d'information, sous l'effet de l'évolution de l'informatique et plus généralement des outils de traitement de l'information, a pris plusieurs formes ces dernières années. Jusqu'à une période récente, ces systèmes d'information désignaient la collecte automatique et les traitements réalisés à des fins de contrôles ou comptables. L'avancée de l'informatisation conduit au développement de grosses bases de données exploitées par des applications d'*aide à la décision*. L'information collectée permet désormais le déploiement de l'intelligence économique dont l'utilisation dépasse dorénavant les domaines du *marketing* et du *management*. Pour toujours améliorer leur productivité, les entreprises sont de plus en plus confrontées à la nécessité de maîtriser cette information et le choix des outils permettant l'analyse et l'exploitation de ces données s'avère primordial.

D'un autre côté, dans l'esprit d'exploiter les données comme on exploite des « mines », la *fouille de données (Data Mining*) met en jeu un processus automatisé d'exploitation des données élémentaires qui s'inscrit lui-même dans un processus plus complexe qui va des *données* à *l'information* et de *l'information* à la *décision*. Autrement dit, la fouille de données est la discipline la plus répandue dans les entreprises soucieuses d'extraire l'information pertinente dissimulée dans leurs bases de données, en vue d'améliorer leurs processus, la gestion de leur relation client et leur maîtrise des risques.

D'une manière générale, la fouille de données est soit *descriptive*, soit *prédictive* : les techniques descriptives (ou exploratoires) visent à mettre en évidence des informations présentes mais noyées dans le volume des données. C'est le cas des techniques de classification automatique des individus (*clustering*) et de la recherche d'associations de produits. Les techniques prédictives (ou explicatives) visent à extrapoler de nouvelles

informations à partir des informations observées (c'est le cas du classement et de la prédiction).

Dans le cadre du processus d'extraction des connaissances à partir des données (ECD) (*Knowledge Discovery in Databases, KDD*) mis à disposition des entreprises et organisations, nous proposons dans cette thèse d'explorer les méthodes adaptées aux données complexes issues du *système d'information hospitalier français,* où de nombreux attributs de nature *classique* et *symbolique* sont à considérer.

Les spécialistes du domaine de la santé le savent : l'information joue un rôle crucial dans le quotidien des établissements de soins. Le *Système d'Information Hospitalier* (SIH) représente la façon dont l'établissement reçoit, traite et stocke l'ensemble des informations nécessaires à la réalisation et l'analyse de son activité. Le SIH peut être assimilé au système nerveux de la structure de soins : il repose sur un environnement matériel et logiciel gérant les informations complexes et utiles à l'établissement de soins. Dans l'objectif d'introduire une dimension médicale dans l'information collectée sur l'activité hospitalière, et d'obtenir de meilleures descriptions et mesures de cette activité, le ministère de Santé français a mis au point, en Juin 1982, le *Programme de Médicalisation des Systèmes d'Information* (PMSI) [Elghazel, 2005].

Dans le cadre du PMSI, tout séjour hospitalier, effectué dans la partie « court séjour » d'un établissement de soins, fait l'objet d'un *Résumé de Sortie Standardisé* (RSS) constitué d'un ou plusieurs *Résumé d'Unité Médicale* (RUM). Le RSS contient un nombre d'informations administratives et médicales sur le séjour du patient. Il décrit les séjours à l'aide variables univaluées, dites à *description classique* (sexe du patient, âge, durée du séjour, date d'entrée, mois de sortie, etc.) mais également de variables multivaluées (pouvant prendre un ensemble de valeurs), comme les diagnostics médicaux, les actes réalisés pendant le séjour, etc., ces variables étant dites à *description symbolique*.

Un algorithme déterministe sous la forme d'un arbre de décision permet ensuite, à partir de ces informations administratives et médicales, de classer chaque séjour dans une classification prédéfinie constituée de *Groupes Homogènes de Malades* (GHM). Les GHM permettent notamment le calcul des points d'Indice Synthétique d'Activité (ISA) pour chaque établissement, chaque GHM pondérant le séjour en fonction de l'échelle nationale des coûts (établie d'après un certain nombre d'établissements pilotes dotés d'une gestion comptable et financière très cadrée). La somme des dépenses est ensuite rapprochée de la somme des points d'activité produits par les établissements de la région, afin de fixer la valeur régionale du point d'activité, en Euros. C'est ainsi qu'est calculée la dotation théorique semestrielle « Médecine, Chirurgie, Obstétrique (MCO) » pour chaque établissement. Au semestre suivant le versement de la dotation théorique sera ajusté des éventuels écarts constatés entre la dotation réelle MCO de l'établissement et la dotation théorique précédente. L'objectif est alors de réduire ces écarts. Cette démarche conduit donc à réformer les modalités d'allocation budgétaire par le biais du PMSI ce qui entraîne

une modulation importante des budgets sur la base de la classification en GHM. Il apparait donc que cette classification joue un rôle stratégique pour les établissements de santé. Les GHM sont réévalués tous les deux ans environ par l'ATIH (Agence Technique pour l'Information Hospitalière) en fonction d'analyses statistiques variées et des observations des établissements. Des tables de correspondances (entre anciennes et nouvelles classes) sont établies qui permettent de pouvoir analyser l'activité des établissements sur plusieurs années.

Néanmoins, malgré les améliorations successives apportées par les institutions, les établissements de soins (publics et privés) ont tendance à réfuter l'adéquation de la classification en GHM à leurs besoins, et pointent notamment la forte hétérogénéité des classes proposées. En effet, la catégorisation des séjours en GHM engendre une asymétrie d'information qui résulte de la diversité des pathologies et des prises en charge dans un même GHM [Quantin *et al.*, 1999]. On observe que cette hétérogénéité intra-GHM peut conduire certains établissements de soins à adopter un comportement opportuniste. Par exemple, sélectionner les activités rentables au détriment des patients ayant des pathologies moins « lucratives ». Ou bien encore calculer son budget théorique en n'acceptant que les patients dont le coût de traitement sera inférieur à la rémunération du GHM. D'autre part, dès que la prise en charge d'une pathologie nécessite un traitement échelonné sur plusieurs séances (pathologies chroniques), l'établissement de santé va peut-être chercher à gonfler artificiellement son activité en multipliant les ré-hospitalisations et donc les séjours d'un même patient, alors que cette situation devrait théoriquement être codée en « pathologie nécessitant plusieurs séances ». Enfin dans ce contexte de GHM rémunéré « au forfait », le risque qu'un établissement opportuniste voire malhonnête décide de revoir à la baisse la qualité des soins pour mieux maîtriser ses coûts, n'est pas nul.

## 1.2. Objectifs et contributions

Les données médico-économiques du PMSI sont jusqu'à présent utilisées à des fins d'allocation de ressources dans les établissements publics et privés et non à l'évaluation de la qualité des soins. En effet, le PMSI s'intéresse à ce qui est effectivement réalisé et non à ce qui devrait être fait. Cependant, dans l'esprit d'une aide au pilotage stratégique des établissements de soins, le PMSI peut présenter le point de départ d'un système d'information adapté au suivi des pathologies et à leur prise en charge. On peut également le considérer comme un outil de gestion interne pouvant jouer un rôle incitatif à l'évaluation de la qualité des soins, exploitant réellement les informations médico-économiques qu'il comporte.

Dans le cadre de cette thèse, nous proposons d'associer les objectifs économiques et médicaux du PMSI afin de les intégrer dans un même processus d'analyse et d'aide à la décision hospitalière. Nous mettons en place deux propositions qui consistent à :

1. Analyser l'hétérogénéité des GHM actuels et fournir une *typologie plus fine des séjours* qui répondrait aux problèmes de non représentation des GHMs et de limiter les comportements opportunistes des établissements. Pour faciliter l'évolution de ces nouveaux groupes de maladies, nous avons proposé une version incrémentale de la classification, qui permet d'automatiquement mettre à jour la typologie en fonction des arrivées de nouveaux séjours hospitaliers, sans pour autant relancer toute la classification. Pour cet objectif, nous avons développé une nouvelle approche de *classification automatique* qui mixe approche de clustering et théorie des graphes (technique de *b-coloration de graphes)* et développé *l'approche incrémentale* associée.

2. La deuxième proposition concerne la mise au point de méthodes d'analyse du PMSI pour *l'aide au pilotage stratégique* des établissements de soins (publics et privés). Il s'agit cette fois d'analyser les trajectoires hospitalières dans le but de fournir aux professionnels de santé un outil d'aide à la décision qui leur permettra d'anticiper leurs activités et de mieux organiser leurs ressources. Une trajectoire est ici composée d'une séquence d'épisodes de soins d'un même patient qui peuvent être effectués dans différents services et/ou établissements de soins.

**Classification automatique des séjours hospitaliers**

La première partie de nos travaux concerne la réalisation d'une nouvelle typologie de séjours hospitaliers. Dans ce cadre, la classification proposée est effectuée a posteriori, sachant que nous ne disposons d'aucune information préalable autre que la description des séjours issue du PMSI. Cette description comporte des variables de nature *classique*, avec par exemple des données quantitatives (âge, durée de séjour,…) et qualitatives (sexe, provenance, sortie, …), mais aussi de nature *symbolique* comme par exemple l'ensemble des actes médicaux et chirurgicaux, et l'ensemble des diagnostics médicaux.

Compte tenu de l'hétérogénéité de ces données, nous avons étudié plusieurs distances adaptées aux données symboliques et nous avons sélectionné celle qui convenait le mieux aux données de l'application.

Par la suite nous avons cherché à construire une partition fine de l'ensemble d'individus (les séjours dans notre cas) en classes, dont les membres vérifient un certain nombre de conditions nécessaires et suffisantes portant sur la *cohésion intraclasse* et la *distance interclasse*. C'est dans ce cadre que nous avons défini une nouvelle méthode de *classification automatique sur tableau de dissimilarités,* qui permet à la fois de garantir l'obtention de groupes homogènes et bien distincts, mais aussi d'identifier le ou les individus représentatifs des classes obtenues. Cette méthode de clustering repose sur une technique de la théorie des graphes baptisée *b-coloration*. Cette technique consiste à

colorer tous les sommets d'un graphe *G* avec un nombre maximum de couleurs (les couleurs représentant ici les classes) tel que : (1) deux sommets adjacents ne portent pas la même couleur (*coloration propre*), sachant qu'avec un tableau de dissimilarités deux sommets dits adjacents sont en fait "dissimilaires" par rapport à un seuil fixé ; (2) pour toute couleur *c*, il existe au moins un sommet s, appelé *sommet dominant*, coloré de cette couleur et adjacent à toutes les autres couleurs. Ce sommet est le reflet des propriétés de la classe mais garantit aussi une séparation nette de la classe vis-à-vis des autres classes de la partition. Cette méthode de coloration a notamment fait ses preuves pour le routage de l'information dans les réseaux distribués.

Suite à cette méthode de classification automatique, nous avons proposé une extension qui concerne *l'apprentissage automatique*, dans l'objectif d'affecter un nouvel individu (ou un groupe d'individus, i.e. de séjours patients) à la classe la plus adéquate ou de retirer un individu, sans relancer toute la classification sur l'ensemble des données. Nous avons donc proposé un *algorithme incrémental de classification automatique* se basant sur la connaissance des dissimilarités entre les individus pris deux à deux. Une première partition est ainsi construite à l'aide de la méthode proposée précédemment puis la partition est mise à jour en fonction des nouvelles arrivées et suppressions de données. La problématique de l'apprentissage automatique est fondée pour un certains nombre d'applications comme l'analyse de l'activité boursière, l'analyse sémantique du contenu web, etc. C'est pourquoi nous l'avons proposée même si le mode d'application de l'approche incrémentale pour la classification des séjours hospitaliers reste à expliciter avec nos partenaires.

**Classification et prévision des trajectoires hospitalières**

La deuxième partie de nos travaux concerne la mise au point des méthodes d'analyse du PMSI dans l'esprit d'une aide au pilotage stratégique des établissements de soins (publics et privés). Nous avons ainsi été amenés à :

- Proposer des méthodologies permettant de retracer la trajectoire d'un patient au sein même du système de soins, en définissant une typologie de trajectoires.
- Développer un outil d'aide à la décision pour la prévision des trajectoires patient. L'objectif est ici d'être capable d'anticiper l'activité de l'établissement de santé pour lui permettre autant que possible de mieux organiser ses ressources (humaines et matérielles) et d'évaluer les coûts du séjour dès l'arrivée du patient.

Cette partie trouve ses fondements scientifiques dans le domaine de la fouille de données et l'extraction des connaissances issues de bases de données de grande taille et séquentielles (temporelles). Nous avons cherché à mettre en relation statistique les

différents enregistrements PMSI en s'appuyant sur des mesures de similarité entre les séquences afin d'identifier des groupes, et définir ainsi une typologie de trajectoires. Notre proposition s'est basée dans un premier temps sur l'introduction d'une dissimilarité sur données séquentielles qui permet à la fois de comparer des séquences de longueurs différentes et aussi de tenir compte de la dilatation dans les séquences. Par la suite, un modèle de classification automatique des données séquentielles a été défini par le couplage de l'approche de classification automatique définie précédemment, avec un modèle probabiliste de chaînes de Markov.

L'approche par *b-coloration* fournit des classes de trajectoires homogènes caractérisées chacune par un ensemble de *trajectoires types* (*profils de patient*), alors que les chaînes de Markov permettent d'interpréter les classes au moyen de modèles probabilistes. Ces derniers fournissent un *cadre automatique de prévision* des trajectoires patient : pour un patient ayant eu différents épisodes de soins, il s'agit dans un premier temps d'identifier la classe de trajectoires dont il se rapproche le plus. Dans un deuxième temps, si c'est nécessaire, nous pouvons prévoir quel sera le séjour suivant le plus probable, et d'en estimer les caractéristiques principales (type de séjour -classe-, diagnostic médical principal, mode de sortie, actes à subir, etc.). Chaque propriété est affectée des probabilités obtenues d'après le modèle de Markov établi pour la classe de trajectoires.

En théorie, ce modèle permettrait aux établissements de soins de connaître a priori le niveau de ressources nécessaires pour fournir au patient les soins nécessaires, et de planifier les ressources nécessaires en conséquence. Ce système peut également fournir une aide à la décision pour les cas « limites » (*border line*), pour lesquels le responsable médical peut souhaiter appuyer sa décision avec une analyse statistique de l'historique des patients (patients de tous les établissements de la région).

Pour pouvoir tester l'ensemble de ces contributions, nous avons développé une plateforme logicielle facile à utiliser par un novice, et adaptée au contexte de l'analyse des trajectoires patients. La plateforme regroupe différents modules permettant de classer un ensemble de trajectoires patients, modéliser les classes de la typologie obtenue à l'aide des chaînes de Markov et évaluer la pertinence des résultats obtenus. La plateforme a été conçue de manière générique et peut accepter tout type d'arborescences (diagnostics, actes, GHMs) puisque nous avons utilisé XML pour les données d'entrée ainsi que pour l'archivage des différents paramètres des chaînes de Markov, afin de gérer les opérations d'ajout, de suppression des trajectoires et l'affichage des statistiques liées au groupes de trajectoires obtenus. Les résultats sur les trajectoires sont ainsi exploitables par différents types d'applications (HTML, tableurs, etc.).

## 1.3. Organisation de la thèse

Le mémoire de thèse est organisé comme suit. Le chapitre 2 présente un état de l'art général des travaux qui traitent du problème de *classification automatique* (*clustering*). A ce titre, nous introduisons les différents types de données pouvant être soumis à une approche de classification automatique. Nous insistons particulièrement sur le problème de la définition d'un *indice de ressemblance* dans le but de pouvoir classer des individus de description *hétérogène et complexe*. Nous présentons également une synthèse des méthodes classiques de classification automatique fournies par la littérature. L'objectif recherché est d'introduire et de positionner nos contributions au regard de l'existant. Pour pouvoir apprécier la justesse des résultats obtenus par la classification, nous exposons par ailleurs les *indices de performance* traditionnellement utilisés dans le problème d'évaluation du clustering.

Le chapitre 3 est consacré à la présentation détaillée de notre approche de classification automatique par *b-coloration de graphes*. Nous évoquons les objectifs et les motivations qui nous ont poussés à choisir cette technique particulière de la théorie des graphes. Puis nous exposons en détail les formalismes utilisés et les différentes étapes de l'approche. Nous présentons également l'extension proposée pour l'apprentissage automatique.. Ce chapitre inclut aussi les processus expérimentaux effectués qui permettent d'évaluer la performance de nos différentes propositions.

Dans le chapitre 4, nous abordons le problème *d'analyse de données séquentielles*. Nous nous intéressons plus particulièrement à la tâche de *classification automatique des séquences temporelles* en évoquant les approches les plus répandues dans la littérature (les *approches par proximité* et les *approches par modèles de mélange*). Pour chacune de ces approches, nous exposons les principes fondamentaux ainsi que les forces et faiblesses de la méthode. Cette synthèse permettra de présenter des approches alternatives de classification et de positionner notre contribution par couplage de l'approche de *classification par b-coloration* et du *modèle de mélange markovien*.

Le chapitre 5 présente un nouveau cadre générique d'analyse de données séquentielles, basé sur la synthèse précédente, *i.e.* l'exploitation du couplage de l'approche de la *classification par b-coloration* et du *mélange markovien*. Nous exposons ensuite l'application de ce cadre d'analyse à un jeu de données issues du domaine médical. Nous utilisons pour cela des jeux de trajectoires de patients (succession de séjours hospitaliers) extraits d'une base de données PMSI-2003 fournie par l'Agence Régionale d'Hospitalisation Rhône-Alpes (ARH-RA). Ce chapitre présente également l'outil d'aide à la décision hospitalière que nous avons mis en place pour concrétiser, sur un plan technique, nos contributions théoriques.

Enfin, le chapitre 6 conclut ce mémoire en présentant un bilan général de l'ensemble de nos contributions et en évoquant de nouvelles perspectives de recherche.

# PARTIE 1

## CLASSIFICATION DES DONNEES HETEROGENES

## APPLICATION AUX SEJOURS HOSPITALIERS

# ÉTAT DE L'ART : LA CLASSIFICATION AUTOMATIQUE

## Résumé

*Dans ce chapitre, nous abordons une recherche bibliographique et nous exposons une synthèse des travaux qui traitent le problème de classification automatique (clustering). Notre synthèse est organisée en quatre parties.*

*Dans une première section, nous introduisons les différents types de données pouvant être soumis à une approche de classification automatique. Dans la deuxième section nous nous intéresserons plus particulièrement au problème de la définition d'un indice de ressemblance dans le but de pouvoir classer des individus de description hétérogène. La troisième section vise à présenter les méthodes classiques de classification automatique (hiérarchiques, par partitionnement et celles fondées sur la théorie des graphes) rencontrées dans la littérature. L'objectif recherché est d'introduire et de positionner la méthode basée sur la b-coloration de graphe proposée dans cette thèse par rapport à ces approches. Enfin, le problème d'évaluation de la qualité d'une partition obtenue par une approche de classification automatique fera l'objet de la dernière section de ce chapitre.*

## Sommaire

# Chapitre 2

# État de l'art : La classification automatique

*" Le collectionneur est un créateur essayant de bâtir un tout cohérent et par là capable de faire œuvre originale ! "*

Michel Melot, "Extrait de la Revue de l'Art"

## 2.1. Introduction

Avec l'expansion des outils informatiques au cours des dernières décennies, un nombre considérable de données de la vie de tous les jours est mis à disposition des analystes. L'archivage de ces données crée la mémoire de notre société, mais la mémoire ne vaut pas grande chose sans l'intelligence. L'intelligence nous permet de mettre de l'ordre dans notre mémoire en observant des formes, en établissant des règles, en trouvant des nouvelles idées qui méritent d'être essayer et en faisant des prédictions.

La *fouille de données* (Data Mining en anglais) [Jambu, 1999; Tufféry, 2005], ou encore *analyse intelligente des données*, désigne l'ensemble de méthodes destinées à l'exploration et l'analyse des données informatiques, de façon automatique ou semi-automatique, en vue de détecter dans ces données des règles, des tendances inconnues ou cachées, des structures particulières restituant l'essentiel de l'information pertinente. La fouille de données permet d'ajouter de l'intelligence aux archives de données pour être capable ensuite de décider.

D'une manière générale, la fouille de données traduit l'ensemble des techniques descriptives (ou exploratoires) visant à mettre en évidence des informations présentes mais cachées par le volume des données (classification automatique), ou aussi prédictives cherchant à extrapoler de nouvelles informations à partir des informations présentes dans les données (classement, prédiction).

Dans le cadre de cette thèse, nous abordons le problème de la classification automatique. La classification automatique ou *analyse de clusters* (*clustering* en anglais), est la tâche qui segmente une population hétérogène en un certain nombre de groupes, plus homogènes, appelés *clusters*.

Dans le clustering, contrairement au *classement*, il n'y a pas de variable cible privilégiée et on ne dispose d'aucune autre information préalable que la description des données en une liste de variables communes. C'est pour cette raison que nous dirons que le clustering est une tâche d'apprentissage "non supervisée" où les enregistrements sont regroupés en fonction de leur similitude de manière à vérifier les deux propriétés suivantes :

- Les objets d'une même classe (*cluster* ou *groupe*) sont aussi similaires que possible : c'est l'*homogénéité intraclasse* (cohésion) qui traduit cette caractéristique.
- Les objets appartenant à des classes différentes sont aussi différents que possible : la plus grande *hétérogénéité interclasse* (séparation) est recherchée.

A l'issue de l'étape de classification, il convient de déterminer quelle signification, si elle existe, doit être accordée aux clusters résultants. Ainsi, utilisé en marketing, le clustering peut indiquer différents profils constituant une clientèle et ainsi la détection de ces classes permet à l'entreprise d'élaborer une offre et une communication spécifique à chacune d'elles. Dans le domaine médical, le clustering permet de déterminer des groupes de patients susceptibles d'être soumis à des suivis thérapeutiques déterminés, chaque classe regroupant les patients réagissant identiquement.

Comme pour toutes les méthodes de fouilles, les données non structurées ne sont pas directement analysables par les techniques de classification automatique : elles se présentent le plus souvent sous la forme d'un tableau rectangulaire avec en lignes les individus (objets, entités, instances, etc.) et en colonnes des variables (attributs, caractéristiques, etc.) (*c.f.* Tableau 2.1).

| | | ***Variables*** | | | | |
|---|---|---|---|---|---|---|
| | ***Y*** / ***X*** | $Y_1$ | ⋯ | $Y_j$ | ⋯ | $Y_m$ |
| ***Individus*** | $X_1$ | | | | | |
| | ⋮ | | | | | |
| | $X_2$ | | | $x_{ij}$ | | |
| | ⋮ | | | | | |
| | $X_n$ | | | | | |

**Tableau 2.1 - Tableau de données**

Dans certains cas, l'utilisateur dispose au départ d'une matrice de ressemblances (similarités, dissimilarités ou distances) entre les objets à classer, sinon il la construit à partir de ses données. Ces mesures de ressemblance entre objets dépendent de la nature des variables mesurées.

La suite de ce chapitre s'énonce comme suit. Nous rappelons dans une première partie (&2.2) les différents types de données pouvant être soumis à une technique de classification automatique. Nous présentons ensuite les mesures de ressemblances généralement utilisées dans les techniques de classification (&2.3). La partie principale de ce chapitre (&2.4) sera consacrée à la synthèse bibliographique des différentes méthodes classiques de classification automatique. Enfin, le problème d'évaluation de la qualité de la partition obtenue par clustering terminera ce chapitre (&2.5).

## 2.2. Les différents types de données rencontrés

La classification intervient sur des données qui résultent d'une suite de choix qui vont influencer les résultats de l'analyse. Classiquement, les données sont décrites dans un tableau individus-variables par une valeur unique. On parlera alors de *« tableau de descriptions univaluées ou classiques »*. Dans les applications réelles, où le grand souci est de prendre en compte la variabilité et la richesse d'informations au sein des données, il est courant d'avoir affaire à des données complexes et hétérogènes (ou mixtes). Ce qui se traduit par le fait que chaque case du tableau de descriptions peut contenir non seulement une valeur unique mais également un ensemble de valeurs, un intervalle de valeurs ou une distribution sur un ensemble de valeurs. On dira alors que la classification va porter sur un *« tableau de descriptions symboliques ».*

### 2.2.1. Description classique d'une variable

Classiquement, une variable $Y_h$ est définie par une application :

$$Y_h : X \rightarrow O_h$$
$$X_i \in X \rightarrow Y_h(X_i)$$

où $X = \{X_1, X_2, \dots, X_n\}$ est l'ensemble des individus. L'ensemble d'arrivée $O_h$ est appelée domaine d'observation de la variable $Y_h$. Un individu est alors décrit sur une variable $Y_h$ par une valeur unique de $O_h$.

On distingue schématiquement deux types de variables : les variables *quantitatives* dites aussi *numériques* et les variables *qualitatives* dites aussi *catégorielles*.

#### *2.2.1.1. Les variables quantitatives*

Une variable *quantitative* prend des valeurs ordonnées (comparable par la relation d'ordre ≤) pour lesquelles des opérations arithmétiques telles que différence et moyenne aient un sens. Une variable quantitative peut être *binaire*, *continue* ou *discrète*. Les variables *binaires* ne peuvent prendre que deux valeurs, le plus souvent associées à {0,1}, {absence, présence} ou {succès, échec} (*exemple* : le sexe d'un nouveau né). Les variables *continues* ou *d'échelle* sont les variables dont les valeurs forment un sous-ensemble infini de l'ensemble $\boldsymbol{R}$ des réels (*exemple* : le salaire, le coût du séjour). Les variables *discrètes*

sont celles dont les valeurs forment un sous-ensemble fini ou infini de l'ensemble ***N*** des entiers naturels (*exemples* : le nombre de jours d'hospitalisation, le nombre d'enfants).

#### *2.2.1.2. Les variables qualitatives*

Une variable *qualitative* (ou aussi *catégorielle*) est une donnée dont l'ensemble des valeurs est fini. Elle prend des valeurs symboliques qui désignent en fait des *catégories* ou aussi *modalités* (*exemples* : le code de la ville, la couleur des cheveux). On ne peut effectuer aucune opération arithmétique sur ces variables. Les variables catégorielles sont parfois ordonnées : on parle alors de variables *ordinales* (*exemple* : faible, moyen, fort). Les variables ordinales peuvent être rangées dans la famille des variables discrètes et traitées comme telles. Les variables catégorielles non ordonnées sont dites *nominales* et lorsque ces variables ont pour valeur des textes non codés, écrits en langage naturel, elles sont dites variables *textuelles* (*exemples* : rapports, nom de film).

### 2.2.2. Description symbolique d'une variable

Dans le cadre de l'analyse de données symboliques introduit par Diday en 1991 [Diday, 1991], la définition d'une variable a été étendue afin de pouvoir décrire un individu par des variables $Y_h$ ayant plusieurs modalités du domaine d'observation $O_h$ [Chavent, 1997; El-Golli, 2004]. Le domaine d'arrivée d'une variable $Y_h$ à description symbolique sera alors modifié par rapport à celui d'une variable classique $O_h$. On notera $\Delta_h$ ce nouvel ensemble d'arrivée. La variable $Y_h$ est ainsi définie par l'application suivante :

$$Y_h : X \rightarrow \Delta_h$$
$$X_i \in X \rightarrow Y_h(X_i)$$

Le domaine d'arrivée $\Delta_h$ peut s'écrire à partir du domaine de valeurs élémentaires $O_h$ et nous pouvons distinguer les trois types de domaine $\Delta_k$ suivants :

- $\Delta_h = O_h$. C'est le cas des variables de valeurs uniques classiques présentées dans la section 2.1. On parlera ainsi de variable à *description univaluée*, quantitative ou qualitative. Par exemple $Y_h(X_i) = rectangle$.

- $\Delta_h = P(O_h)$ avec $P(O_h)$ est l'ensemble de parties de $O_h$. C'est le cas d'une variable qualitative qui peut être décrite par plusieurs modalités ou d'une variable quantitative qui être décrite par un intervalle de valeurs. On parlera alors de *description multivaluée*. Par exemple $Y_h(X_i) = \{rectangle, carré\}$.

- $\Delta_h = [0,1]^{O_h}$, l'ensemble des fonctions de $O_h$ dans [0,1]. On parlera alors de *description modale*. Par exemple, $Y_h(X_i)$ est une distribution de probabilité sur l'ensemble de valeurs $\{rectangle, carré\}$.

- Le tableau 1.2 suivant présente des exemples de descriptions multivaluées et modales des variables *salaire* et *forme géométrique*.

| | ***Salaire*** | ***Forme géométrique*** |
|---|---|---|
| ***Multivaluéee*** | [1500,2500] | $\{rectangle, carré\}$ |
| ***Modale*** | Densité de la loi normale $LN(2000, \sigma)$ | $Prob(rectangle) = 0.7$<br>$Prob(carré) = 0.3$<br>$Prob = 0\ ailleurs$ |

**Tableau 2.2 - Exemple de descriptions multivaluées et modales**

### *2.2.2.1. Les variables à descriptions multivaluées*

C'est le cas d'une variable $Y_h$ qui peut être décrite par plusieurs valeurs du domaine d'observation $O_h$.

- Si le domaine d'observation $O_h$ est quantitatif (continu, discret) ou qualitatif ordinal, la description multivaluée de $Y_h$ est un intervalle de valeurs et le domaine d'arrivée $\Delta_h$ de $Y_h$ est l'ensemble des intervalles fermées bornés sur $O_h$. Par exemple, la variable $Y_h = coût\ d'hospitalisation$ pour une intervention sur le rachis peut être $Y_h\ (intervention\ sur\ le\ rachis) = [5161,9236]$.
- Si le domaine d'observation $O_h$ est qualitatif nominal, la description multivaluée de $Y_h$ est un ensemble de valeurs et le domaine d'arrivée $\Delta_h$ de $Y_h$ est l'ensemble de sous-ensembles de $O_h$. Par exemple, la variable $Y_h = traitements\ subis\ au\ cours\ d'une\ hospitalisation$ pour le patient *toto* peut prendre les valeurs $Y_h\ (toto) = \{Uncusectomie, Foraminotomie\}$.

Dans [Chavent, 1997], l'auteur précise qu' « *au niveau sémantique, les descriptions multivaluées permettent de traduire les notions d'imprécision et de variabilité dans la description des individus* ».

Soit l'individu $X_i$ ayant comme description relative à la variable *forme géométrique* l'ensemble de valeurs $Y_h(X_i) = \{rectangle, \text{carré}\}$. Cela peut correspondre à une *imprécision* due à un *doute* : cet objet a une forme *carrée* ou *rectangulaire*. Si pour la variable *coût d'hospitalisation,* $Y_h(intervention\ sur\ le\ rachis) = [5161,9236]$, cela peut correspondre a un *bruit* : le séjour d'hospitalisation pour une hospitalisation sur le rachis peut coûter entre 5161€ et 9236€. Dans le premier cas, la notion de *vraie forme géométrique* n'a pas vraiment de sens car elle dépend du jugement de chacun. Dans le deuxième cas, nous pouvons supposer que le *vrai coût du séjour d'intervention sur le rachis* appartient à l'intervalle [5161,9236].

D'autre part, un intervalle ou un ensemble de valeurs peuvent permettre d'introduire la notion de ***variabilité*** dans une description. Par exemple, l'intervalle [60,195] peut exprimer la variation de la *pression artérielle* d'un patient au cours d'un séjour d'hospitalisation. L'ensemble $\{Uncusectomie, Foraminotomie\}$ peut exprimer la liste de tous les actes médicaux subis par un patient lors d'un séjour d'hospitalisation pour une

intervention sur le rachis. Il s'agit ici de variabilité due au caractère temporel de la variable.

#### *2.2.2.2. Les variables à descriptions modales*

C'est le cas d'une variable $Y_h$ qui peut être décrite par une fonction définie sur le domaine d'observation $O_h$ dans [0,1].

Cette fonction peut être une distribution de probabilité sur $O_h$ ou une fonction d'appartenance d'ensembles flous de $O_h$ . Par exemple, nous pourrons indiquer que le coût d'un séjour d'intervention sur le rachis est uniformément distribué sur l'intervalle [5161, 9236], ou encore normalement distribué autour de la valeur 7198.5. Dans ce cas, le coût du séjour est décrit par la fonction de densité de la loi normale de moyenne 7198.5 et d'écart type $\sigma$.

Contrairement au cas multivalué, où les valeurs prises par une variable traduisent l'imprécision sans donner un degré de certitude quant à ces valeurs, les variables modales permettent de traduire la notion d'imprécision à partir de la notion d'incertitude. Par exemple, pour la variable $Y_h = forme\ géométrique$ où le domaine d'observations $O_h$ est défini par un ensemble de valeurs précises, un degré d'incertitude peut être fourni pour traduire la notion d'imprécision dans la description des données. On pourra dire ainsi que la forme de l'objet est « rectangulaire avec une certitude de $2/3$ » et « carrée avec une certitude de $1/3$ ».

#### *2.2.2.3. Les variables taxonomiques ou structurées*

Les domaines d'observation des variables de classification peuvent être munis parfois de connaissances supplémentaires appelées *connaissances du domaine*. Ces connaissances supplémentaires sont définies dans le cas de descriptions univaluées mais peuvent être prises en compte dans un traitement sur des descriptions multivaluées (*exemple* : dans le calcul de la mesure de ressemblance entre les individus au cours d'un processus de classification automatique).

Il arrive par exemple qu'un expert puisse fournir une structuration des valeurs du domaine d'observation sous la forme d'un arbre ordonné, d'un graphe orienté, etc. Une variable dont le domaine d'observation est représenté par une structure hiérarchique est dite *variable taxonomique* ou encore *structurée* [Ichino and Yaguchi, 1994; Michalski and Stepp, 1983].

La figure 2.1 fournit une partie de la structuration hiérarchique du domaine d'observation de la variable *diagnostic médical*. Cette structuration est appelée *la Classification Internationale des Maladies* (CIM10).

***Remarque 2.1 :*** Dans la suite de ce mémoire, nous limiterons notre étude au cas des variables multivaluées $Y_h$ dont le domaine d'observation $O_h$ est *qualitatif nominal* (*i.e.* le domaine d'arrivée $\Delta_h$ est un ensemble de valeurs).

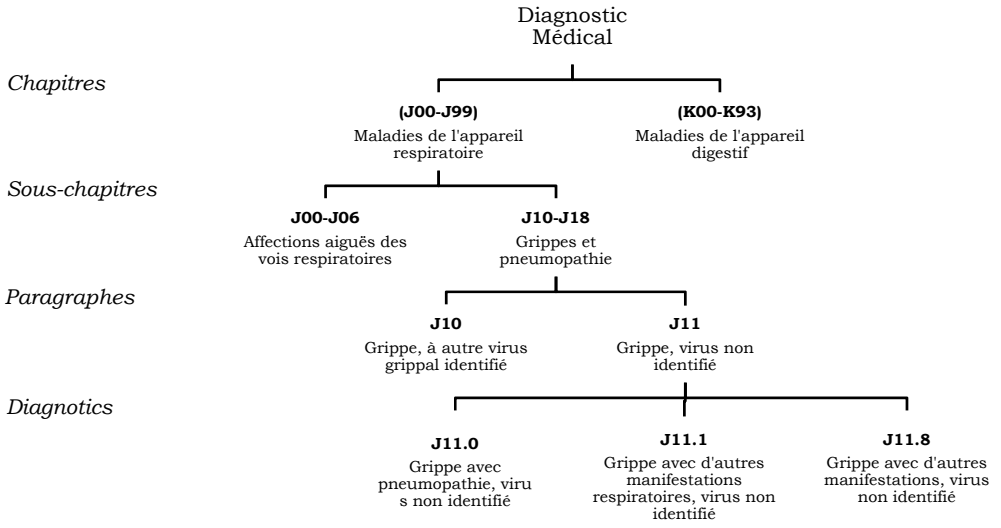



**Figure 2.1 - Partie de la structure hiérarchique sur les modalités de la variable diagnostic médical**

## 2.3. Les mesures de ressemblance

Tout système ayant pour but d'analyser ou d'organiser automatiquement un ensemble de données ou de connaissances doit utiliser, sous une forme ou une autre, un opérateur capable d'évaluer précisément les ressemblances ou les dissemblances qui existent entre ces données.

La notion de ressemblance (ou *proximité*) a fait l'objet d'importantes recherches dans des domaines extrêmement divers. Pour qualifier cet opérateur, plusieurs notions comme la *similarité*, la *dissimilarité* ou la *distance* peuvent être utilisées [Bisson, 2000].

La suite de cette section vise d'abord à définir ces différentes notions (*similarité*, *dissimilarité* et *distance*), les liens qui existent entre elles et leurs différences. Il s'agit ensuite de faire une présentation des mesures de ressemblance définies pour les données selon qu'elles sont de nature *classique* ou *symbolique*. Enfin nous verrons comment un indice de proximité peut être défini entre des individus à *vecteur de descriptions symboliques* [Bock and Diday, 2001; Chavent, 1997; Ichino and Yaguchi, 1994] dans le but de les classer par la suite.

## 2.3.1. Définitions

Nous appelons *similarité* ou *dissimilarité* toute application à valeurs numériques qui permet de mesurer le lien entre les individus d'un même ensemble. Pour une similarité le lien entre deux individus sera d'autant plus fort que sa valeur est grande. Pour une dissimilarité le lien sera d'autant plus fort que sa valeur de la dissimilarité est petite [Celeux *et al.*, 1989].

### *2.3.1.1. Indice de dissimilarité*

Un opérateur de ressemblance $d : X * X \rightarrow R^+$ défini sur l'ensemble d'individus $X = \{X_1, X_2, \dots, X_n\}$ est dit *indice de dissimilarité* (ou *dissimilarité*), s'il vérifie les propriétés suivantes :

1. $\forall\ X_i, X_j \in X;\ d(X_i, X_j) = d(X_j, X_i)$ (*propriété de symétrie*)
2. $\forall\ X_i \in X;\ d(X_i, X_j) \geq d(X_i, X_i) = 0$ (*propriété de positivité*)

### *2.3.1.2. Distance*

Un opérateur de ressemblance $d : X * X \rightarrow R^+$ défini sur l'ensemble d'individus $X = \{X_1, X_2, \dots, X_n\}$ est dit *distance*, s'il vérifie en plus des deux propriétés 1 et 2 les propriétés d'*identité* et d'*inégalité triangulaire* suivantes :

3. $\forall\ X_i, X_j \in X;\ d(X_i, X_j) = 0 \Rightarrow X_i = X_j$ (*propriété de d'identité*)
4. $\forall\ X_i, X_j, X_k \in X;\ d(X_i, X_j) \leq d(X_i, X_k) + d(X_k, X_j)$ (*inégalité triangulaire*)

### *2.3.1.3. Indice de similarité*

Un opérateur de ressemblance $s : X * X \rightarrow [0,1]$ défini sur l'ensemble d'individus $X = \{X_1, X_2, \dots, X_n\}$ est dit *indice de similarité* (ou *similarité*), s'il vérifie en plus de la propriété de symétrie (1) les deux propriétés suivantes :

5. $\forall\ X_i \in X;\ s(X_i, X_j) \geq 0$ (*propriété de positivité*)
6. $\forall\ X_i, X_j \in X\ et\ X_i \neq X_j;\ d(X_i, X_i) = s(X_j, X_j) > s(X_i, X_j)$ (*propriété de maximisation*)

Il convient de noter ici que le passage de l'indice de similarité $s$ à la notion duale d'indice de *dissimilarité* (que nous noterons $d$), est trivial. Etant donné $s_{max}$ la similarité d'un individu avec lui-même ($s_{max} = 1$ dans le cas d'une similarité normalisée), il suffit de poser :

$$\forall\ X_i, X_j \in X; d(X_i, X_j) = s_{max} - s(X_i, X_j) \qquad (2.1)$$

## 2.3.2. Mesure de ressemblance entre individus à descriptions classiques

Le processus de classification vise à structurer les données contenues dans $X = \{X_1, X_2, \dots, X_n\}$ en fonction de leurs ressemblances, sous la forme d'un ensemble de classes à la fois homogènes et contrastées.

L'ensemble d'individus $X$ est décrit généralement sur un ensemble de $m$ variables $Y = \{Y_1, Y_2, \dots, Y_m\}$ définies chacune par :

$$Y_h : X \to \Delta_h$$
$$X_i \in X \to Y_h(X_i)$$

où $\Delta_k$ est le domaine d'arrivée de la variable $Y_h$ .

En conséquence, les données de classification sont décrites dans un tableau Individus-Variables où chaque case du tableau contient la description d'un individu sur une des $m$ variables. Ce tableau Individus-Variables est en général un tableau homogène qui peut être de type *quantitatif* (où toutes les variables sont quantitatives) ou *qualitatif* (où toutes les variables sont qualitatives).

### *2.3.2.1. Tableau de données numériques (continues ou discrètes)*

La distance la plus utilisée pour les données de type quantitatives *continues* ou *discrètes* est la *distance de Minkowski d'ordre* $\propto$ définie dans $R^m$ par :

$$\forall\ X_i, X_j \in X;\ d(X_i, X_j) = \left( \sum_{h=1}^{m} \left| Y_h(X_i) - Y_h(X_j) \right|^{\propto} \right)^{\frac{1}{\propto}} \tag{2.2}$$

où $\propto \geq 1$, avec si :

- $\propto = 1$, $d$ est la distance de *City-block* ou *Manhattan*.

$$d(X_i, X_j) = \sum_{h=1}^{m} \left| Y_h(X_i) - Y_h(X_j) \right| \tag{2.3}$$

- $\propto = 2$, $d$ est la distance *Euclidienne* classique.

$$d(X_i, X_j) = \sqrt{\sum_{h=1}^{m} \left( Y_h(X_i) - Y_h(X_j) \right)^2} \tag{2.4}$$

- $\propto \to +\infty$, $d$ est la distance de *Chebyshev* définie comme suit :

$$d(X_i, X_j) = \max_{1 \leq h \leq m} \left| Y_h(X_i) - Y_h(X_j) \right| \tag{2.5}$$

On a le plus souvent recours à la distance euclidienne mais la distance de Manhattan est aussi parfois utilisée, notamment pour atténuer l'effet de larges différences dues aux points *atypiques* (*aberrants* ou *outliers*) puisque leurs coordonnées ne sont pas élevées au

carré. Il est à noter que dans la plupart des cas, la distance de Manhattan donne des résultats semblables à ceux de la distance euclidienne.

#### *2.3.2.2. Tableau de données binaires*

Les $n$ individus à classer sont décrits par $m$ variables binaires codées 0 ou 1. La ressemblance entre deux individus $X_i$ et $X_j$ se calcule à partir des informations du tableau de contingence 2$x$2 ci-dessous. Un tel tableau permet de compter le nombre de concordances ($a$+$d$) et le nombre de discordances ($b$+$c$) entre les individus.

| | | $X_j$ | |
|---|---|---|---|
| | | **1** | **0** |
| $X_i$ | **1** | $a$ | $b$ |
| | **0** | $c$ | $d$ |

**Tableau 2.3 - Tableau de contingence**

Il convient de noter que le rôle des modalités d'une variable binaire est très important dans le calcul d'une mesure de ressemblance entre les individus. En effet, une variable binaire peut être *symétrique* (les modalités 0 et 1 de cette variable ont la même importance) ou *asymétrique* (les deux modalités n'ont pas la même importance).

Parmi les nombreux indices de similarité (encore appelés *coefficients d'associations*) entre deux individus $X_i$ et $X_j$, les plus connus sont :

- L'indice de Jaccard [Jaccard, 1908] : il est défini pour des variables asymétriques et ne tient pas compte de l'absence conjointe :

$$\forall\ X_i, X_j \in X;\ d(X_i, X_j) = \frac{a}{a+b+c} \tag{2.6}$$

- L'indice de Sokal et Sneath [Sokal and Sneath, 1963] : il ignore également l'absence conjointe mais contrairement à l'indice de Jaccard compte doublement les discordances :

$$\forall\ X_i, X_j \in X;\ d(X_i, X_j) = \frac{a}{a+2*(b+c)} \tag{2.7}$$

- L'indice de Russel et Rao où l'absence conjointe n'est pas considérée comme une similarité :

$$\forall\ X_i, X_j \in X;\ d(X_i, X_j) = \frac{a}{a+b+c+d} \tag{2.8}$$

- L'indice Simple Match de Sokal et Michener [Sokal and Michener, 1958] : est défini pour des variables symétriques et l'absence conjointe est considérée comme une similarité.

$$\forall\ X_i, X_j \in X;\ d(X_i, X_j) = \frac{a+d}{a+b+c+d} \tag{2.9}$$

Nous pouvons aussi utiliser la distance euclidienne pour définir les ressemblances entre individus d'un tableau de données toutes binaires.

Pour une liste plus complète de mesures de similarité à utiliser dans le cas de variables binaires, voir par exemple [Celeux *et al.*, 1989; Nakache and Confais, 2005].

#### *2.3.2.3. Tableau de données ordinales*

Il s'agit d'un tableau de données où les variables qui décrivent les individus sont *qualitatives ordinales*. Les valeurs $Y_h(X_i)$ sont remplacées par leurs rangs $R_h(X_i)$, $R_h(X_i) = 1.,2, \dots, p_h$, où $p_h$ est le nombre de valeurs distinctes de la variable $Y_h$. Ces valeurs sont par la suite transformées en utilisant la formule ci-dessous qui fournit une variation $Z_h(X_i)$ entre 0 et 1 de $Y_h(X_i)$ :

$$\forall\ X_i \in X;\ Z_h(X_i) = \frac{R_h(X_i) - 1}{p_h - 1} \tag{2.10}$$

La distance entre deux individus $X_i, X_j$ est ainsi calculée à partir des variations $Z_h$ considérées comme des données numériques.

#### *2.3.2.4. Tableau de données nominales*

La distance la plus utilisée pour les données de type *qualitative nominale* est la *distance de Hamming*. Etant donné deux individus $X_i, X_j \in X = \{X_1, X_2, \dots, X_n\}$ décrits chacun par $m$ variables nominales, la distance de Hamming entre $X_i$ et $X_j$ est donnée par le nombre de caractéristiques de $X_i$ qui diffèrent de celle de $X_j$ comme suit :

$$\forall\ X_i, X_j \in X;\ d(X_i, X_j) = \sum_{h=1}^{m} Y_h(X_i) \oplus Y_h(X_j)$$

où $$Y_h(X_i) \oplus Y_h(X_j) = \begin{cases} 1, & Y_h(X_i) \neq Y_h(X_j) \\ 0, & sinon \end{cases} \tag{2.11}$$

Une autre distance largement utilisée pour les tableaux de données nominales est la distance de Hamming *normalisée*. Elle est basée sur le calcul du nombre d'appariements entre les individus. Etant donné deux individus $X_i$ et $X_j$ présentant les mêmes modalités pour $p$ parmi les $m$ variables nominales, alors :

$$\forall\ X_i, X_j \in X;\ d(X_i, X_j) = \frac{m-p}{m} \tag{2.12}$$

Pour une liste plus complète de mesures de similarité à utiliser dans le cas de variables qualitatives, le lecteur pourra par exemple consulter [Dubes, 1993].

#### *2.3.2.5. Mesure de ressemblance entre variables aléatoires*

Dans certains cas, l'utilisateur désire analyser les variables à la place des individus. Ceci requière la définition d'un opérateur capable d'évaluer la proximité entre ces variables d'analyse.

L'incertitude sur une variable aléatoire $Y_i$ (respectivement un couple de variables aléatoires $(Y_i, Y_j)$) peut être mesurée par *l'entropie* notée $H(Y_i)$ (respectivement $H(Y_i, Y_j)$).

La quantité notée $I(Y_i : Y_j)$ (équation 2.13), appelée *information mutuelle*, mesure l'information transmise entre $Y_i$ et $Y_j$.

$$I(Y_i : Y_j) = H(Y_i) + H(Y_j) - H(Y_i, Y_j) \tag{2.13}$$

Comme nous pouvons montrer que $I(Y_i: Y_j) \leq H(Y_i, Y_j)$, que l'indépendance entre $Y_i$ et $Y_j$ entraîne $I(Y_i: Y_j) = 0$ et qu'une dépendance déterministe bijective entre $Y_i$ et $Y_j$ entraîne $I(Y_i: Y_j) = H(Y_i, Y_j)$, Dussauchoy a proposé dans [Dussauchoy, 1982] une similarité normée et une dissimilarité entre variables aléatoires qui s'écrivent comme suit:

$$s(Y_i, Y_j) = \frac{I(Y_i : Y_j)}{H(Y_i, Y_j)} \tag{2.14}$$

$$d(Y_i, Y_j) = \frac{H(Y_i, Y_j) - I(Y_i : Y_j)}{H(Y_i, Y_j)} \tag{2.15}$$

Il a également généralisé cette notion de dissimilarité pour mesurer la dissemblance entre deux vecteurs aléatoires et a appliqué ces idées à la décomposition de systèmes complexes modélisés à l'aide de vecteurs aléatoires [Dussauchoy, 1982].

### 2.3.3. Mesure de ressemblance entre individus à descriptions symboliques

La présentation de la section précédente illustre la diversité des mesures existantes et l'importance du choix de la distance ou de la similarité dans le processus de classification automatique pour ne pas trop influencer son déroulement. Le choix de la dissimilarité/similarité est facilité lorsqu'on est en présence d'un type unique de données. Néanmoins dans les applications réelles, il est courant d'avoir affaire à des données de différents types dites *hétérogènes* ou *mixtes*. Ce qui se traduit par le fait que la classification porte sur un *« tableau de données symboliques ou complexes»* contenant à la fois des variables *univaluées* (*quantitatives* ou *qualitatives*) et *multivaluées* (nous nous limiterons ici à un *ensemble de modalités*).

Dans ce cas, les mesures de proximités usuelles entre deux individus ne sont pas directement applicables, il faut donc développer de nouvelles approches. Il existe principalement deux stratégies pour répondre à ce problème :

***1ère Approche* (*homogénéisation du tableau de descriptions*)** : il s'agit de transformer les variables pour les homogénéiser (afin qu'elles aient en fin le même type), puis utiliser une fonction de comparaison globale qui tient compte de toutes les variables afin de calculer un indice de proximité entre les individus.

*Dans le cas classique*, où les individus sont décrits à la fois par des variables *univaluées* (quantitatives et qualitatives), plusieurs opérations de conversion qui permettent de passer d'un type à un autre sont définies dans la littérature, mais ces transformations induisent une perte d'information et une distorsion dans les résultats. Par exemple, une variable nominale peut être transformée en autant de variables binaires que de modalités qu'elle présente. Une variable quantitative peut être transformée en une variable qualitative ordinale en effectuant un découpage du domaine d'observation ($R$ ou $N$) à l'aide de bornes définies par l'utilisateur puis en affectant à chaque individu le numéro de la classe du découpage à laquelle il appartient.

De telles transformations étant réalisées, le tableau de données devient homogène et nous retombons dans le cadre classique de la section 2.3.2 précédente.

*Dans le cas symbolique*, où les individus sont décrits à la fois par des variables *univaluées* (quantitatives et qualitatives) et *multivaluées*, il ne s'agit pas de transformer les descriptions multivaluées en une modalité unique afin de ne pas perdre l'information contenue dans ces descriptions, mais plutôt de passer d'une description univaluée à une description multivaluée. Par exemple une valeur $v$ peut être vu comme l'intervalle $[v, v]$ dans le cas d'une variable quantitative ou l'ensemble $\{v\}$ dans le cas d'une variable qualitative [Chavent, 1997]. Ces différentes transformations permettent d'obtenir un tableau homogène où toutes les descriptions sont multivaluées. Il reste maintenant à définir un indice de proximité global qui tient compte de l'ensemble de ces descriptions afin de mesurer la ressemblance entre les individus (paragraphes 2.3.3.1 et 2.3.3.2).

***2ème Approche* (*agrégation des comparaisons sur les variables*)** : il s'agit de l'approche généralement utilisée pour comparer deux individus à descriptions symboliques. Cette approche ne nécessite aucune transformation préalable du tableau de données. Son principe est le suivant :

- Définir pour chaque variable $Y_h$ de l'ensemble des $m$ variables caractéristiques $\mathrm{Y} = \{Y_1, Y_2, \dots, Y_m\}$ une fonction de comparaison $g_h$ .
- Utiliser la fonction d'agrégation proposée par Ichino et Yaguchi dans [Ichino and Yaguchi, 1994] qui repose sur la *métrique de Minkowsky* pour combiner les différentes comparaisons obtenues sur chaque variable dans une même mesure de ressemblance.

C'est à cette approche que nous nous intéressons dans cette thèse.

#### 2.3.3.1. Fonctions de comparaison entre descriptions univaluées

Une variable à description univaluée peut être quantitative ou qualitative. Les fonctions de comparaison élémentaires les plus utilisées pour des données univaluées sont les suivantes :

- Pour une variable quantitative $Y_h$:

$$\forall\ X_i, X_j \in X;\ g_h\left(Y_h(X_i), Y_h(X_j)\right) = \left|Y_h(X_i) - Y_h(X_j)\right| \tag{2.16}$$

Cette fonction de comparaison peut être normalisée en la divisant par un coefficient de normalisation $m_h$ calculant l'écart maximal de la variable $Y_h$. Il est définit comme suit :

$$m_h = \max_{X_i \in X} Y_h(X_i) - \min_{X_i \in X} Y_h(X_i) \tag{2.17}$$

- Pour une variable qualitative $Y_h$:

$$\forall\ X_i, X_j \in X;\ g_h\left(Y_h(X_i), Y_h(X_j)\right) = \begin{cases} 0, & Y_h(X_i) = Y_h(X_j) \\ 1, & Y_h(X_i) \neq Y_h(X_j) \end{cases} \tag{2.18}$$

#### 2.3.3.2. Fonctions de comparaison entre descriptions multivaluées

Comme précisé auparavant, nous nous intéressons dans cette thèse au cas des variables multivaluées dont le domaine d'observation $O$ est nominal (*i.e.* domaine d'arrivée $\Delta$ est un ensemble de modalités). Compte tenu de la description de ces variables, nous allons maintenant définir quelques fonctions de comparaisons entre ensembles. La littérature regorge de définitions de distances entre descriptions.

Pour une liste des fonctions de comparaison à utiliser dans le cas de variables à descriptions multivaluées (intervalle de valeurs) et modales, voir les articles [Bock, 2001; Esposito *et al.*, 2001; Malerba *et al.*, 2001].

##### a. La distance de Jaccard

Une liste d'extensions d'indices de ressemblances entre données binaires a été proposée dans [De-Carvalho, 1994] afin de calculer la similarité entre des ensembles de valeurs. Le plus utilisé et le plus simple à calculer est *l'indice de Jaccard*. Etant données les deux descriptions $Y_h(X_i)$ et $Y_h(X_j)$ correspondantes à la variable $Y_h$ pour respectivement les deux individus $X_i$ et $X_j$, l'indice de similarité de Jaccard est défini dans le cas symbolique comme suit :

$$\forall\ X_i, X_j \in X;\ g_h\left(Y_k(X_i), Y_h(X_j)\right) = \frac{\left|Y_h(X_i) \cap Y_h(X_j)\right|}{\left|Y_h(X_i) \cup Y_h(X_j)\right|} \tag{2.19}$$

où || est le *cardinal*.

L'idée sous-jacente est que deux descriptions se ressemblent d'autant plus que leur intersection est importante et leur union réduite. Cette similarité permet ainsi d'obtenir la

valeur maximale de 1 lorsque les deux ensembles sont identiques, et la valeur minimale (0), s'ils sont totalement disjoints.

### b. La dissimilarité de Gowda et Diday

Gowda et Diday [Gowda and Diday, 1991] définissent une dissimilarité entre deux ensembles de modalités prenant en compte les ressemblances au niveau de leurs *valeurs communes* et aussi les ressemblances de *cardinal*, en se basant sur les deux fonctions suivantes :

$\forall\ X_i, X_j \in X;$

$$D_s\left(Y_h(X_i), Y_h(X_j)\right) = \frac{\left|l_i - l_j\right|}{l_s} \qquad (2.20)$$

$$D_c\left(Y_h(X_i), Y_h(X_j)\right) = \frac{l_i + l_j - 2 * \left|Y_h(X_i) \cap Y_h(X_j)\right|}{l_s}$$

où

- $l_i$ est le nombre de valeurs dans $Y_h(X_i) = |Y_h(X_i)|$.
- $l_j$ est le nombre de valeurs dans $Y_h(X_j) = \left|Y_h(X_j)\right|$.
- $l_s$ est le nombre de valeurs dans $Y_h(X_i) \cup Y_h(X_j) = |Y_h(X_i)| + \left|Y_h(X_j)\right| - \left|Y_h(X_i) \cap Y_h(X_j)\right|$.

Au moment où la fonction $D_s$ compare le *cardinal* des deux ensembles de valeurs, la fonction $D_c$ compare leurs *contenus*.

Se basant sur ces deux fonctions, la dissimilarité de Gowda et Diday entre les deux descriptions $Y_h(X_i)$ et $Y_h(X_j)$ est donnée par la formule suivante :

$$\forall\ X_i, X_j \in X;\ g_h\left(Y_h(X_i), Y_h(X_j)\right) = D_s\left(Y_h(X_i), Y_h(X_j)\right) + D_c\left(Y_h(X_i), Y_h(X_j)\right) \qquad (2.21)$$

Gowda et diday ont également défini une fonction de comparaison pour mesurer la ressemblance entre deux intervalles de valeurs [Gowda and Diday, 1991].

### c. La fonction de comparaison d'Ichino et Yaguchi

Ichino et Yaguchi [Ichino and Yaguchi, 1994] proposent une mesure de dissimilarité entre deux sous-domaines $Y_h(X_i)$ et $Y_h(X_j)$ d'une variable $Y_h$ comme suit :

$$\forall\ X_i, X_j \in X;\ g_h\left(Y_h(X_i), Y_h(X_j)\right) = \left|Y_h(X_i) \oplus Y_h(X_j)\right| - \left|Y_h(X_i) \cap Y_h(X_j)\right| + \gamma\left(2\left|Y_h(X_i) \cap Y_h(X_j)\right| - \left|Y_h(X_i)\right| - \left|Y_h(X_j)\right|\right) \quad (2.22)$$

où

- $|Y_h(X_i)|$ est le *cardinal* de l'ensemble $Y_h(X_i)$.
- $0 \leq \gamma \leq 0.5$. Le choix de la valeur $\gamma$ est laissé à l'utilisateur sachant qu'Ichino et Yaguchi préconisent de prendre $\gamma = 0.5$ pour le cas général.

$$\forall\ X_i, X_j \in X;\ g_h\left(Y_h(X_i), Y_h(X_j)\right) = \left|Y_h(X_i) \oplus Y_h(X_j)\right| - 0.5\left(\left|Y_h(X_i)\right| + \left|Y_h(X_j)\right|\right) \quad (2.23)$$

- $\oplus$ est l'opérateur d'union jointe définie par Ichino & Yaguchi pour tous les types de descriptions symboliques. Lorsque $O_h$ est nominal (*i.e.* $Y_h(X_i)$ est un *ensemble de modalités*), cas auquel nous nous intéressons dans cette thèse, l'opérateur d'union jointe est équivalent à l'union ensembliste $\cup$ $\left(i.e.\ Y_h(X_i) \oplus Y_h(X_j) = Y_h(X_i) \cup Y_h(X_j)\right)$.

Ichino et Yaguchi proposent également une normalisation de leur mesure de dissimilarité en la divisant par un coefficient de normalisation $m_h$ défini par le nombre de modalités possibles dans le domaine d'observation $O_h$ de la variable $Y_h$.

A l'aide de la définition 2.1 suivante, Ichino et Yaguchi ont d'autre part défini un opérateur d'union jointe pour le cas des variables taxonomiques [Ichino and Yaguchi, 1994].

***Définition 2.1*** *Etant donnée une variable $Y_h$ dont le domaine d'observation $O_h$ est nominal et structuré dans une hiérarchie, soient $Y_h(X_i)$ et $Y_h(X_j)$ deux descriptions de $P(O_h)$ relatives respectivement aux individus $X_i$ et $X_j$. $Y_h(X_i)$ et $Y_h(X_j)$ sont donc deux ensembles de $O_h$ et $N(Y_h(X_i))$ est la classe la plus fine dans la hiérarchie qui contient tous les éléments de $Y_h(X_i)$. L'union $Y_h(X_i) \oplus Y_h(X_j)$ est définie comme suit :*

- *si* $N(Y_h(X_i)) = N\left(Y_h(X_j)\right)$ alors $Y_h(X_i) \oplus Y_h(X_j) = Y_h(X_i) \cup Y_h(X_j)$.
- *sinon* $Y_h(X_i) \oplus Y_h(X_j) =$ *les valeurs des feuilles situées sous le nœud* $N\left(Y_h(X_i) \cup Y_h(X_j)\right)$ *dans la structure hiérarchique.*

***Exemple :*** Toujours dans le cadre médical du *Programme de Médicalisation des Systèmes d'Information* (PMSI), prenons comme exemple la variable $Y_1 = Actes\ médicaux$ qui définit, pour tout individu (patient), les actes (traitements) médicaux qu'il subit pendant son d'hospitalisation. Le domaine d'observation de cette variable est une structure hiérarchique (arbre ordonné) appelée *Catalogue des Actes Médicaux* (CdAM). Cette classification des actes médicaux comporte 6 niveaux de découpage (champ, chapitre, sous-chapitre, paragraphe, sous-paragraphe, rubrique). Chaque niveau peut être considéré

comme le regroupement de tous les actes élémentaires situés dans les niveaux inférieurs de la hiérarchie.

Les tableaux 2.4 et 2.5 ci-dessous présentent respectivement la description de 7 actes médicaux ainsi que leur emplacement dans la structure hiérarchique du CdAM de la figure 2.2.

| ***Acte*** | ***Libell***é |
|---|---|
| W846 | Autogreffe corticale et/ou spongieuse nécessitant un deuxième site opératoire : Pied |
| W847 | Mise en place d'un appareil de stimulation externe pour consolidation : Pied |
| W848 | Ostéotomie simple Médio-Tarsienne, y compris Tarsectomie pour pied creux |
| W849 | Autres interventions osseuses : Pied |
| W850 | Arthrodèse Sous-Astragalienne |
| W851 | Double Arthrodèse Sous-Astragalienne et Médio-Tarsienne ou équivalent |
| W852 | Arthrotomie simple : Avant-Pied |

**Tableau 2.4 - Exemple d'actes médicaux et description**

| ***Acte*** | ***Champ*** | ***Chapitre*** | ***Sous − Chapitre*** | ***Paragraphe*** | ***Sous − Paragraphe*** | ***Rubrique*** |
|---|---|---|---|---|---|---|
| W846 | 1 | 14 | 1 | 23 | 2 | 4 |
| W847 | 1 | 14 | 1 | 23 | 2 | 4 |
| W848 | 1 | 14 | 1 | 23 | 2 | 5 |
| W849 | 1 | 14 | 1 | 23 | 2 | 7 |
| W850 | 1 | 14 | 1 | 23 | 3 | 0 |
| W851 | 1 | 14 | 1 | 23 | 3 | 0 |
| W852 | 1 | 14 | 1 | 23 | 4 | 0 |

**Tableau 2.5 - Positions des actes médicaux dans la structure hiérarchique du CdAM**

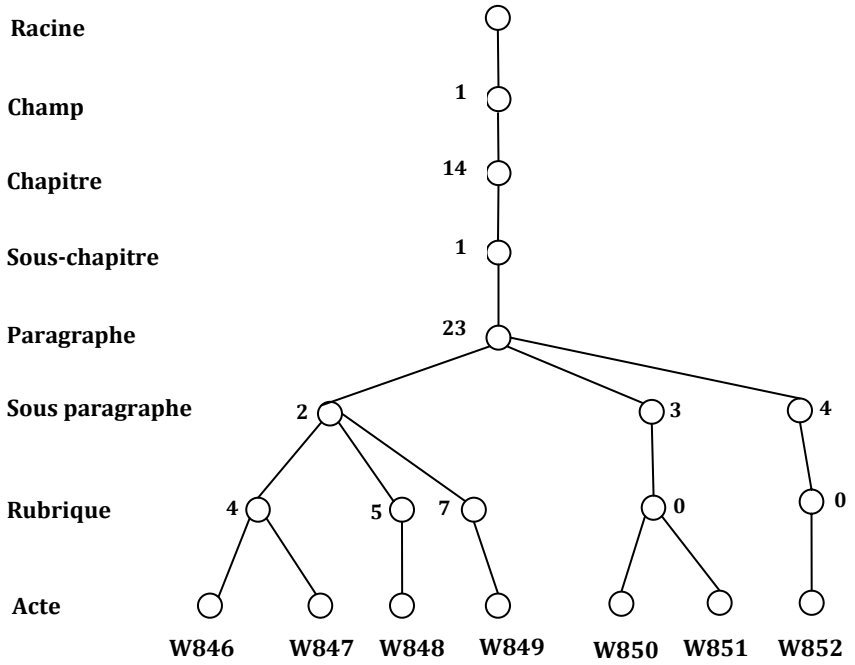



**Figure 2.2 - Structure hiérarchique du catalogue CdAM associé à la variable « Actes médicaux»**

Prenons par exemple le cas de deux séjours hospitaliers (individus) $X_1$ et $X_2$ ayant respectivement comme descriptions relatives à la variable $Y_1 = Actes\ médicaux$ les deux ensembles suivants : $Y_1(X_1) = \{W846, W849\}$ et $Y_1(X_2) = \{W847, W848\}$. La classe la plus fine contenant tous les éléments de $Y_1(X_1)$ est le sous-paragraphe 1.14.1.23.2 $\left(i.e.\ N\left(Y_1(X_1)\right) = 1.14.1.23.2\right)$ et c'est également la classe la plus fine contenant tous les éléments de $Y_1(X_2)$ $\left(i.e.\ N\left(Y_1(X_2)\right) = 1.14.1.23.2\right)$. Ainsi $N\left(Y_1(X_1)\right) = N\left(Y_1(X_2)\right)$, et donc la fusion $Y_1(X_1) \oplus Y_1(X_2) = \{W846, W849, W847, W848\}$ . Nous en déduisons que la dissimilarité de Ichino entre ces deux descriptions est : $g_1\left(Y_1(X_1), Y_1(X_2)\right) = 4 - 0 + 0.5(2 * 0 - 2 - 2) = 2$.

***Remarque 2.2 :*** Nous avons défini dans cette section des éléments permettant de comparer les descriptions multivaluée (ensemble de valeurs) de deux individus sur une variable. Pour cela, nous avons présenté un ensemble de fonctions $g_h$ appelées *fonctions de comparaison* de descriptions entre ensembles de modalités. Nous observons toutefois qu'avec ces fonctions, la dissimilarité entre les individus ne diminue que grâce aux éléments qui sont communs. En effet, elles ne prennent pas en compte la ressemblance sémantique qui peut exister entre leurs éléments et qui peut être cachée par exemple dans la structure hiérarchique du domaine d'observation de la variable de description. Ceci n'est pas satisfaisant car cela ne prend pas en compte la ressemblance sémantique entre deux ensembles, que nous cherchions justement à obtenir.

La structure hiérarchique du catalogue CdAM fournit un très bon exemple pour illustrer notre propos. Comme nous l'avons expliqué précédemment, la structure du CdAM

est une hiérarchie matérialisée par un découpage en champs (ALPHA, BÊTA, GAMMA, etc…). Les champs sont découpés eux-mêmes en chapitres (par exemple, pour le champ ALPHA : Système nerveux, Ophtalmologie, ORL, etc...), lesquels se subdivisent en sous-chapitres, qui à leur tour sont découpés en paragraphes, puis en sous-paragraphes, et enfin en rubriques. Une rubrique est le niveau de regroupement le plus fin d'un nombre plus ou moins important d'actes élémentaires, voisins par leurs caractéristiques systémiques, anatomiques, techniques, etc. Nous en déduisons que deux patients qui ont subi des actes appartenant à la même rubrique peuvent être considérés médicalement proches.

Soient trois séjours hospitaliers $X_1$,$X_2$ et $X_3$ ayant respectivement comme descriptions relatives à la variable $Y_1 = Actes\ médicaux$ les ensembles suivants : $Y_1(X_1) = \{W846, W850\}$, $Y_1(X_2) = \{W847, W851\}$ et $Y_1(X_3) = \{W848, W852\}$. La classe la plus fine contenant tous les éléments de $Y_1(X_1)$ est le paragraphe 1.14.1.23 et c'est également celle qui contient tous les éléments de $Y_1(X_2)$ et de $Y_1(X_3)$ . Nous en déduisons une *dissimilarité d'Ichino et Yaguchi* entre $Y_1(X_1)$ et $Y_1(X_2)$ de l'ordre de 2 $\left(g_1\left(Y_1(X_1), Y_1(X_2)\right) = 4 - 0 + 0.5(2 * 0 - 2 - 2) = 2\right)$ et la même valeur pour $Y_1(X_1)$ et $Y_1(X_3)$. Ainsi, la fonction de comparaison d'Ichino situe $Y_1(X_1)$ à la même distance de $Y_1(X_2)$ et de $Y_1(X_3)$, bien que les deux premières descriptions soient clairement plus proches, vu les caractéristiques voisines de leurs éléments ($W846$ et $W847$ et également $W850$ et $W851$ se trouvent sous la même rubrique dans la structure du CdAM). Ce qui n'est pas le cas pour $Y_1(X_1)$ et $Y_1(X_3)$. Cet état de fait n'étant pas pris ne compte lors du calcul de la ressemblance, il ne permet pas de baisser la dissimilarité entre les individus.

D'autres fonctions de comparaison entre ensembles de modalités ont été définies dans la littérature et qui permettent de lever ce problème sémantique. Parmi eux, nous trouvons la *dissimilarité de Hausdorff*.

***Définition 2.2*** *La distance de Hausdorff* [Rote, 1991] *mesure l'éloignement de deux descriptions ensemblistes $Y_h(X_i)$ et $Y_h\left(X_j\right)$ d'un espace métrique de la façon suivante :*

$$\forall\ X_i, X_j \in X;\ g_h\left(Y_h(X_i), Y_h\left(X_j\right)\right) = max\left\{\sup_{a \in Y_h(X_i)}\left\{\inf_{b \in Y_h\left(X_j\right)}(d(a,b))\right\}, \sup_{b \in Y_h\left(X_j\right)}\left\{\inf_{a \in Y_h(X_i)}(d(a,b))\right\}\right\} \quad (2.24)$$

où $d(a, b)$ est la dissimilarité entre deux éléments $a$ et $b$.

La dissimilarité de Hausdorff définit ainsi une mesure de dissemblance qui est une vraie distance si la dissimilarité $d$ entre les éléments est elle-même une distance. Le principal problème de cette distance est qu'elle ne prend pas compte la structure globale des ensembles de valeurs à comparer $\left(i.e.\ Y_h(X_i)\right.$ et $\left.Y_h\left(X_j\right)\right)$. En effet, cette mesure est très sensible aux éléments extrêmes. Par exemple, les ensembles {1, 2, 3, 4, 5} et {1} sont à

la même distance de {6} par la distance de Hausdorff (avec $d(a,b) = |a-b|$). Nous verrons dans un *chapitre ultérieur* comment nous modifions cette distance afin de coupler les deux aspects *structurel* et *sémantique* dans le calcul de proximité entre individus. Ceci donnera notamment une faible valeur de distance pour des ensembles *proches* mais non identiques.

#### *2.3.3.3. Vers une mesure de ressemblance entre vecteurs de descriptions symboliques*

Dans le cas d'individus décrits par un ensemble de variables, mesurer la ressemblance entre deux individus revient à mesurer la ressemblance entre leurs vecteurs de description. Pour comparer deux vecteurs de description, on procède souvent par comparaison des descriptions variable par variable, puis par agrégation de ces comparaisons. Ichino et Yaguchi [Ichino and Yaguchi, 1994] ont ainsi proposé une fonction d'agrégation s'inspirant de la *distance de Minkowsky,* pour généraliser la comparaison de deux descriptions sur une variable aux vecteurs de descriptions symboliques sur $m$ variables.

Soient les deux individus $X_i, X_j \in X = \{X_1, X_2, \dots, X_n\}$ de vecteurs de descriptions respectifs $Y(X_i) = \{Y_1(X_i), Y_2(X_i), \dots, Y_m(X_i)\}$ et $Y(X_j) = \{Y_1(X_j), Y_2(X_j), \dots, Y_m(X_j)\}$, Ichino et Yaguchi proposent un indice de proximité entre $X_i$ et $X_j$ comme suit :

$$\forall\ X_i, X_j \in X;\ d(X_i, X_j) = \left( \sum_{h=1}^{m} \left( g_h\left(Y_h(X_i), Y_h(X_j)\right)\right)^{\propto} \right)^{\frac{1}{\propto}} \tag{2.25}$$

où $\propto \geq 1$, avec pour :

- $\propto = 1$, $d$ est la distance de *City-block* ou *Manhattan*.

$$d(X_i, X_j) = \sum_{h=1}^{m} g_h\left(Y_h(X_i), Y_h(X_j)\right) \tag{2.26}$$

- $\propto = 2$, $d$ est la distance *Euclidienne* classique.

$$d(X_i, X_j) = \sqrt{\sum_{h=1}^{m} \left( g_h\left(Y_h(X_i), Y_h(X_j)\right)\right)^2} \tag{2.27}$$

- $\propto \to +\infty$, $d$ est la distance de *Chebyshev* définie comme suit :

$$d(X_i, X_j) = \max_{1 \leq h \leq m} \left\{ g_h\left(Y_h(X_i), Y_h(X_j)\right)\right\} \tag{2.28}$$

- $\propto$ *quelconque*, $d$ est la distance de *Minkowsky généralisée d'ordre* $\propto$.

Cette mesure de dissemblance peut être une dissimilarité ou une distance selon les cas suivants :

- si les fonctions de comparaison $g_h$ sont toutes des *dissimilarités*, la mesure de ressemblance *d* est une *dissimilarité*.

- si les fonctions de comparaison $g_h$ sont toutes des *distances*, la mesure de ressemblance *d* est une *distance*.

Lorsque toutes les fonctions de comparaison $g_h$ sont normalisées, cette mesure convient tout à fait aux problèmes d'analyse de données hétérogènes car, non seulement elle gère les descriptions qualitatives, quantitatives et multivaluées, mais de plus, elle permet d'atténuer les effets de pondération dus aux écarts qui peuvent figurer entre les unités de mesures choisies pour les différentes variables. Elle donnera ainsi le même poids à toutes les variables pour le calcul de la ressemblance.

## 2.4. Les techniques classiques de classification automatique

La classification automatique a été utilisée depuis longue date dans des contextes variés par des chercheurs de différentes disciplines, en tant que processus d'analyse exploratoire de données. Elle fut la cause d'innombrables travaux théoriques et applicatifs et est encore le foyer de journaux spécialisés et de communautés actives à travers le monde. Les études actuelles démontrent encore le vif intérêt de la classification automatique, tant dans les façons possibles de l'appliquer que pour l'améliorer.

Les méthodes de classification automatique d'un ensemble d'individus peuvent être divisées en deux grandes familles : les *approches hiérarchiques* et les *approches par partitionnement* [Berkhin, 2002; Jain *et al.*, 1999]. Les *approches hiérarchiques*, qui produisent une séquence de partitions emboîtées d'hétérogénéités croissantes de la plus fine à la plus grossière, conduisent à des résultats sous forme d'*arbre hiérarchique indicé* connu aussi sous le nom de *dendrogramme*, qui visualise ce système de classes organisées par inclusion. Contrairement aux approches hiérarchiques, les *approches par partitionnement* cherchent la meilleure partition en *k* classes disjointes des données, le nombre de classes (*clusters* ou *groupes*) *k* étant fixé a priori. Les *approches par partitionnement* utilisent un processus itératif fonction du nombre *k* qui consiste à affecter chaque individu à la classe la plus proche au sens d'une distance -ou d'un indice de similarité- en optimisant une certaine *fonction objectif*.

### 2.4.1. Les approches hiérarchiques de classification

La construction d'une classification hiérarchique peut se faire de deux façons : pour la première, à partir d'une matrice symétrique des similarités entre les individus, un *algorithme agglomératif* forme initialement de petites classes ne comprenant que des individus très semblables, puis, à partir de celles-ci, il construit des classes de moins en moins homogènes, jusqu'à obtenir la classe entière. Ce mode de construction est appelé

*Classification Ascendante Hiérarchique* (*CAH*). Le second mode de construction d'une classification hiérarchique inverse le processus précédent. Il repose sur un *algorithme divisif* muni d'un critère de division d'un sous-ensemble de variables, et qui procède par dichotomies successives de l'ensemble des individus tout entier, jusqu'à un niveau qui vérifient certaines règles d'arrêt et dont les éléments constituent une partition de l'ensemble des individus à classer. Ce mode de construction s'appelle la *Classification Descendante Hiérarchique*. Un autre mode de construction des classes a été proposé par Diday [Diday, 1986], comme une généralisation des modèles hiérarchiques, est appelé la *classification pyramidale*. Comme les hiérarchies, les représentations pyramidales sont des ensembles de *parties* appelées aussi *classes* ou *paliers* de l'ensemble des individus à classer. Cependant, la représentation pyramidale constitue une structure plus complexe des données. En effet, contrairement au cas hiérarchique classique, deux classes de la pyramide peuvent avoir une intersection non vide et ainsi certains individus à classer, peuvent appartenir à deux classes qui ne sont pas *emboîtées* l'une dans l'autre (*classes empiétantes*). La hiérarchie obtenue dans ce cas est dite *hiérarchie de recouvrement* (ou *pyramide*). Dans le cadre de cette thèse, nous nous intéressons seulement aux cas où les individus appartiennent à une seule classe (partition). Ainsi, nous détaillons dans la suite de ce paragraphe les approches hiérarchiques classiques conduisant à une hiérarchie de partitions. Cependant, nous trouvons dans l'article de Bertrand et Diday [Bertrand and Diday, 1990] des détails concernant les approches pyramidales.

#### *2.4.1.1. La Classification Ascendante Hiérarchique*

Le schéma d'un algorithme de Classification Ascendante Hiérarchique (CAH) est le suivant :

1. Les classes initiales sont les individus eux-mêmes.
2. On calcule les distances entre les classes.
3. Les deux classes les plus proches sont fusionnées et remplacées par une seule.
4. Le processus reprend en 2 jusqu'à n'avoir plus qu'une seule classe, qui contient toutes les observations.

Un algorithme *agglomératif* fonctionne donc en recherchant à chaque étape les classes les plus proches pour les fusionner, et l'étape la plus importante dans l'algorithme réside dans le choix de la distance entre deux classes. Les algorithmes les plus classiques définissent la distance entre deux classes à partir de la mesure de dissimilarité entre les objets constituant chaque groupe. De nombreuses distances sont ainsi possibles :

- L'algorithme *Single-linkage* où la distance entre deux clusters est représentée par la distance minimum entre toutes les paires de données entre les deux clusters (paire composée d'un élément de chaque cluster), nous parlons alors de *saut minimum*. Le point fort de cette approche est qu'elle sait très bien détecter les classes allongées, mais son point faible est qu'elle est sensible à

*l'effet de chaîne*[1] [Tufféry, 2005] et donc moins adaptées pour détecter les classes sphériques.

- L'algorithme *Complete-linkage* où la distance entre deux clusters est représentée par la distance maximum entre toutes les paires de données des deux clusters, nous parlons alors de *saut maximum* ou du *critère de diamètre*. Par définition cette approche est très sensible aux points aberrants et donc elle est peu utilisée [Tufféry, 2005].

- L'algorithme *Average-linkage* propose de calculer la distance entre deux clusters en prenant la valeur moyenne des distances entre tous les couples d'objets des deux clusters. Nous parlons aussi de *saut moyen*. Cette approche tend à produire des classes de même variance.

- L'algorithme *Centroid-linkage* (ou *saut barycentrique*) définit, quant à lui, la distance entre deux clusters comme la distance entre leur *centre de gravité*. Une telle méthode est plus robuste aux points aberrants. Toutefois, elle est limitée aux données quantitatives numériques pour lesquelles le calcul du centre de gravité est possible.

Des exemples classiques de tels algorithmes sont contenus dans [Kaufman and Rousseeuw, 1990]. Ces méthodes classiques sont intéressantes dans le sens où la plupart d'elles sont fondées sur un lien métrique qui les rend applicables à tout type de données dès lors que l'on est capable de construire une matrice de ressemblances entre les individus à classer. Néanmoins elles présentent également des inconvénients non négligeables. La complexité algorithmique de ces classifications n'est pas linéaire, car, pour passer de *k+1* classes à $k$ classes, il faut calculer $(k+1)k/2$ distances, réunir les deux classes les plus proches, puis recalculer les distances, avant de recommencer. Si $n$ est le nombre d'individus à classer, la complexité de ces algorithmes est au moins en $O(n^2)$, voire $O(n^3)$ pour les méthodes les plus simples. D'autre part, comme nous l'avons remarqué auparavant, la forme des classes obtenues est très dépendante des distances d'agrégation utilisées : *le saut minimum* donne plutôt des classes allongées tandis que le *saut maximum* donne plutôt des classes sphériques. En général ces méthodes conduisent le plus souvent à des partitions formées de classes de forme convexe, de taille et densité sensiblement égales, sans tenir compte éventuellement des points aberrants. Or dans plusieurs applications, les classes révélées sont de forme arbitraire.

Plus récemment, de nouvelles méthodes de classification hiérarchique ont été développées, afin d'éviter la majorité des problèmes décrits ci-dessus, et notamment pour fournir des partitions en classes de forme et taille arbitraires. Parmi celles-ci citons CURE, ROCK, BIRCH et CHAMELEON, que nous présentons ci-après.

[1] Nous appelons effet de chaîne lorsque deux points très éloignés l'un de l'autre mais reliés par une suite de points très proches les uns des autres sont rassemblés dans la même classe.

### a. CURE (Clustering Using REpresntatives)

Dans la méthode CURE (Clustering Using REpresntatives), proposée par Guha et *al.* [Guha *et al.,* 1998], un échantillon représentatif de taille $s$ de l'ensemble initial est utilisé pour réduire à la fois, la complexité algorithmique et la mémoire nécessaire. Cet échantillon est divisé en $p$ sous-ensembles de taille $s/p$, qui sont individuellement, regroupés en $s/pq$ sous-classes. Après élimination des classes à faible effectif et les points aberrants, plusieurs points sont déterminés pour représenter chaque sous-classe. Ces points représentatifs sont bien répartis à l'intérieur de leurs sous-classes en subissant une homothétie d'un facteur $\alpha$ vers le centre de la classe. La considération de plusieurs points pour représenter une classe et l'utilisation de ce facteur $\alpha$ offrent à l'algorithme respectivement un meilleur ajustement à la géométrie de formes arbitraires et une maîtrise des points aberrants. Dans une deuxième étape, les différentes sous-classes sont agrégées hiérarchiquement en utilisant comme distance entre deux sous-classes, la plus petite distance entre un représentant de la première et un représentant de la deuxième et ce, jusqu'à obtenir le nombre $k$ de classes requis. Une fois les $k$ classes étant retenues, l'algorithme procède à une classification de l'ensemble total des individus en utilisant les $c$ points représentant ces classes. Chaque individu est affecté à la classe possédant le représentant qui lui est le plus proche.

### b. ROCK (Robust Clustering for Categorical Data)

L'algorithme ROCK (Robust Clustering for Categorical Data) a été aussi développé par Guha et *al.* [Guha *et al.,* 2000], pour la classification des données qualitatives. D'une façon analogue à CURE, cet algorithme utilise un échantillon représentatif de l'ensemble initial des données et procède à une classification hiérarchique de cet échantillon jusqu'à atteindre le nombre de classes souhaitées. Cependant, il se diffère de CURE, dans la manière dont il procède pour traiter les données catégorielles et définir ainsi un critère d'agrégation des classes. Il se base ainsi sur le concept de *voisinage* entre deux individus et *d'inter-connexité* entre deux classes pour mesurer une similarité d'agrégation. Plus précisément, deux individus sont dits *voisins* si leur similarité dépasse un certain seuil $\theta$ fixé par l'utilisateur et deux classes seront fusionnées si leur similarité, mesurée par le nombre d'individus des deux classes qui ont des voisins communs, est la plus grande.

### c. CHAMELEON (Hierarchical Clustering Algorithm Using Dynamic Modeling)

CHAMELEON [Karypis *et al.*, 1999] est un algorithme de classification hiérarchique *dynamique* qui partage certaines notions des algorithmes CURE et ROCK, qui utilisent quant à eux, un modèle statique pour déterminer les classes les plus semblables à fusionner. CHAMELEON utilise un graphe de partitionnement des *k-plus proches voisins* dont chaque sommet représente un individu à classer et est relié à ses *k-plus proches voisins*. D'autre part, tout arc est valué par la similarité entre les individus des sommets connectés. Cet algorithme procède en deux étapes : dans la première, un algorithme de

partitionnement de graphes permet d'obtenir un nombre important de sous-classes de petite dimension. Ces sous-classes sont alors fusionnées hiérarchiquement en se basant sur des mesures *d'interconnexité* et de *proximité relatives* entre deux sous-classes. Les liaisons déterminées à partir du graphe de partitionnement des *k-plus proches voisins*, estiment à la fois les densités *intraclasses* et *interclasses* et fournissent ainsi une bonne mesure de proximité entre les sous-classes. Cet algorithme aboutit à des classes de formes et de densités variées. Il présente également l'avantage de ne nécessiter que la définition d'une mesure de similarité pour fonctionner, ce qui le rend applicable à tout type de données. Cependant, la mesure de proximité entre classes utilisée dans le processus de fusion est sensible au bruit et aux points aberrants.

#### *d. BRICH (Balanced Iterative Reducing and Clustering using Hierarchies)*

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) [Zhang *et al.*, 1996] est un algorithme de classification hiérarchique applicable à des ensembles de données de taille importante avec une complexité très faible. Il permet de regrouper les individus en deux étapes principales :

1. La première étape construit le cœur de l'algorithme BIRCH. Dans cette phase, l'algorithme construit dynamiquement (*i.e.* d'une manière incrémentale) et en utilisant certains paramètres de contrôle à définir a priori, un arbre appelé CF (*Clustering Feature*). Les feuilles de cet arbre correspondent à des *micro-classes* caractérisées chacune par un vecteur d'informations qui est ensuite utilisé à l'étape suivante de l'algorithme. Ce sont ces vecteurs CF qui apparaissent dans les nœuds de l'arbre et non pas les individus des sous-classes eux-mêmes, qui ne sont donc exploitées qu'une seule fois durant le processus de classification. C'est pourquoi BIRCH peut traiter un grand volume de données en utilisant une mémoire limitée.

2. A l'issue de l'étape de création des caractéristiques et après avoir éliminé les points aberrants, la deuxième phase de l'algorithme consiste à construire un arbre hiérarchique agglomératif à partir des segments terminaux de l'arbre CF, en utilisant un critère d'agrégation basé sur les résumés de ces segments. On notera que d'autres algorithmes de classification peuvent d'ailleurs être utilisés dans cette phase.

Les quatre approches évoluées de classification ascendante hiérarchique que nous avons détaillées ci-dessus ont été proposées pour remédier aux problèmes associés aux approches classiques, et fournir ainsi des partitions en classes de forme et de taille arbitraires. Cependant, la fiabilité et l'efficacité de ces approches reposent fortement sur le choix de *l'échantillon représentatif* (pour CURE et ROCK) et des *paramètres de contrôles* (pour BIRCH et CHAMELEON), et ce choix n'est pas toujours évident par rapport aux données à classer. Une aide à la décision est ainsi parfois nécessaire qui peut s'appuyer par

exemple sur une démarche plus globale basée sur l'évaluation des partitions obtenues en fonction des choix.

#### *2.4.1.2. La Classification Descendante Hiérarchique*

Les méthodes de classification descendante hiérarchique sont itératives et procèdent à chaque itération au choix du segment de l'arbre hiérarchique à diviser, et au partitionnement de ce segment. La différence entre les méthodes divisives, développées jusqu'à présent dans la littérature, figure dans les critères qu'elles utilisent pour choisir le segment à diviser ainsi dans la manière dont elles divisent le segment. Le choix de tels critères dépend généralement de la nature des variables caractérisant les individus à classer.

Nous verrons dans ce qui suit deux types d'approches divisives : les *approches monothétiques* et les *approches conceptuelles*. Nous trouvons dans la thèse de [Chavent, 1997] et les papiers [Chavant, 1998; Chavent *et al.*, 1999] des détails concernant ce mode de construction de classification hiérarchique ainsi que des références de plusieurs méthodes divisives.

##### *a. Les méthodes monothétiques*

Les méthodes nomothétiques divisent un segment (classe $C$) de l'arbre hiérarchique en deux sous-segments (sous-classes $C_1$ et $C_2$) en fonction d'une variable et de deux groupes de valeurs de cette variable.

Si la variable est quantitative le segment est divisé suivant la réponse à la question de la forme "$valeur\ de\ la\ variable\ \leq c\ ?$". Le premier sous-segment $C_1$ contient les individus pour lesquels la valeur de la variable est inférieure ou égale à $c$, et l'autre $C_2$ les individus pour lesquels la valeur de la variable est strictement supérieure à $c$. Si la variable est qualitative, un individu est affecté au premier sous-segment $C_1$ si sa description pour cette variable appartient à un premier groupe de modalités, sinon il est affecté au deuxième sous-segment $C_2$.

La stratégie utilisée par ces méthodes pour choisir la *variable de division* (parmi celles caractérisant les individus) ainsi que la *valeur de coupure* ($c$ pour les variables quantitatives, et les groupes de modalités pour les variables qualitatives) repose sur l'optimisation d'un critère d'évaluation bien déterminée (par exemple le diamètre d'une partition donné par la plus grande dissimilarité entre deux individus d'une même classe : ainsi nous choisissons la classe et la coupure qui fournissent une partition de petit diamètre).

M. Chavant a proposé dans [Chavent, 1997] une méthode monothétique de classification hiérarchique définie pour tout type de variables quantitatives ou qualitatives, possédant éventuellement des descriptions symboliques. Cette méthode

cherche à chaque étape à diviser la classe qui fournit une nouvelle partition optimisant un critère mathématique *d'inertie intraclasse* défini pour le cas de descriptions symboliques.

#### *b. La classification conceptuelle*

Une approche de classification conceptuelle a été introduite par [Fisher, 1987] comme une méthode divisive de classification automatique sur des données qualitatives. Elle repose sur les deux notions de *Partition Utility* (CU) et *Category Utility* (CU). La fonction PU mesure la qualité d'une partition en $k$ classes de l'ensemble d'individus $X = \{X_1, X_2, \dots, X_n\}$, sur lesquels sont relevées les mesures de $m$ variables qualitatives $Y_h = \{1,2, \dots, m\}$ présentant $m_h$ modalités chacune. La *Partition Utility* représente de fait un compromis entre la similarité *interclasse* et *intraclasse*. En maximisant PU, nous maximisons à la fois la probabilité que deux individus d'une même classe aient des modalités en commun et la probabilité que deux individus de classes différentes aient des modalités différentes. A partir de cette mesure PU, les auteurs ont également défini la fonction CU pour une partition en $k$ classes comme étant l'accroissement du nombre attendu de modalités des variables correctement prédites, par rapport au nombre de modalités prédites correctes sans connaissance de la partition.

A partir de ces deux notions, la stratégie de l'approche conceptuelle pour diviser un segment de l'arbre hiérarchique est la suivante : chacune des variables qualitatives $Y_h$ fournit une partition en $m_h$ classes de ce segment et donc une *Partition Utility* et une *Category Utility*. Le segment est ainsi divisé suivant les modalités de la variable avec le CU maximal. A la fin de l'exécution de cet algorithme, chaque classe de la partition finale obtenue est munie d'une définition conceptuelle qui résume ses éléments.

Les deux notions PU et CU ont d'autre part été étendues pour traiter des variables mixtes quantitatives et qualitatives. Aussi une extension de cette méthode a été proposée dans [Fisher, 1987] pour construire une classification hiérarchique sur un ensemble d'individus présentés séquentiellement. Cette nouvelle approche est appelée COBWEB. Elle ne suit pas une stratégie divisive pour classer les individus mais elle construit dynamiquement le dendrogramme en classant les individus un à un à l'aide d'un algorithme incrémental utilisant des opérations d'insertion, division, fusion et création pour faire la mise à jour de la partition.

### *2.4.1.3. Une approche symbolique de classification ascendante hiérarchique*

En 2003, une approche symbolique de classification ascendante hiérarchique a été proposée par Mali et Mitra [Mali and Mitra, 2003]. Elle suit le même principe de fonctionnement que les approches classiques mais se diffère par le critère d'agrégation qu'elle utilise. En effet, elle définit la distance entre deux classes $C_i$ et $C_j$ comme suit :

$$d_{agrégation}(C_i, C_j) = \frac{\sum_{u=1}^{|C_i|}\sum_{q=1}^{|C_j|} d(X_u, X_q)}{|C_i||C_j|}\left(\frac{|C_i|.|C_j|}{|C_i| + |C_j|}\right)^{\frac{1}{2}} \quad (2.29)$$

où $d$ représente la mesure de dissimilarité de *Gowda et Diday* définie sur l'ensemble d'individus $X$ (section 2.3.3.2) et $|C_i|$ le cardinal (nombre d'individus) de la classe $C_i$.

On observera notamment que le terme de pondération utilisée par cette distance prend une valeur de $\sqrt{50}$ (pour $|C_i| = |C_j| = 100$), de $101/100$ (pour $|C_i| = 1$ et $|C_j| = 100$) et de $\sqrt{0.5}$ (pour $|C_i| = |C_j| = 1$). La distance d'agrégation aura donc de grandes valeurs pour des clusters de large taille et de faibles valeurs pour des clusters de petite taille. En conséquence, l'approche de classification hiérarchique tend à favoriser le fusionnement des classes singletons, ou des petites et grandes classes, au détriment du fusionnement des classes de tailles moyennes.

#### *2.4.1.4. Conclusion*

Les deux modes de construction (*agglomératif* et *divisif*) aboutissent à une classification hiérarchique indicée qui n'est pas forcément la même. Une fois qu'elle est obtenue, il peut être intéressant d'analyser et d'interpréter cette classification, afin de choisir la partition idéale et de fournir ensuite une représentation pour chaque groupe (par exemple par son centre).

Bien que les méthodes hiérarchiques représentent la famille principale des techniques de classification et qui ont été appliquées avec succès dans plusieurs domaines, elles souffrent d'une faiblesse qui réside dans leur critère de partitionnement qui n'est pas global, mais dépend des classes déjà obtenues précédemment. En effet, les opérations de fusions/divisions des classes se déroulent sans jamais remettre en cause les associations déjà constituées, ce qui peut conduire à des classes peu représentatives (notamment en présence de données aberrantes) [NG and Han, 2002]. Pour les cas agglomératif par exemple, deux individus placés dans des classes différentes ne sont jamais plus comparés, et deux individus placés dans une même classe ne peuvent plus être séparés. En d'autres termes, la classification obtenue en *k* classes n'est jamais la meilleure possible (l'optimale), mais seulement la meilleure entre celles obtenues en fusionnant des classes d'une classification en *k+1* classes.

### 2.4.2. Les approches par partitionnement

Les approches de classification par partitionnement permettent de subdiviser l'ensemble des individus en un certain nombre de classes en employant une stratégie d'optimisation itérative dont le principe général est de générer une partition initiale, puis de chercher à l'améliorer en réattribuant les données d'une classe à l'autre. Il n'est bien entendu pas souhaitable d'énumérer toutes les partitions possibles. Ces algorithmes recherchent donc des maxima locaux en optimisant une *fonction objectif* traduisant le fait

que les individus doivent être *similaires* au sein d'une même classe, et *dissimilaires* d'une classe à une autre. Les classes de la partition finale, prises deux à deux, sont d'intersection vide et chacune est représentée par un *noyau* (un ou des individus de la population, ou un point de l'espace).

Les algorithmes de partitionnement sont divisés en trois grandes sous-familles : les méthodes des *k-moyennes* (*k-means*), les méthodes des *k-médianes* (*k-medoids*) et les méthodes des *nuées dynamiques*, selon la définition des représentants des classes. Les deux premières familles de méthodes tendent à construire des classes de forme convexe sans tenir compte réellement des points aberrants. Dans ces familles d'algorithmes, ceux les plus couramment utilisés sont : les *centres mobiles* (*k-means*), PAM, CLARA et CLARANS. D'autre part, ces deux familles d'approches s'énoncent comme des variantes (cas particuliers) de la méthode des nuées dynamiques qui fournit quant à elle une variété de modes de représentation des classes (appelé noyaux), selon l'objectif d'analyse souhaité .

#### *2.4.2.1. Les méthodes des k-moyennes*

La méthode des centres mobiles due à Forgy [Forgy, 1965] est la plus classique et celle qui reste très utilisée. Elle procède comme suit : dans une première étape, elle consiste à tirer aléatoirement $k$ individus de la population. Ces individus représentent les centres provisoires des $k$ classes qui formeront la partition initiale. Ensuite, les autres individus sont regroupés autour de ces $k$ centres en affectant chacun d'eux au centre le plus proche. L'étape suivante consiste à recalculer les $k$ nouveaux centres (dites aussi *centroïdes* ou *centres de gravité*) des $k$ classes, sachant qu'un centre n'est pas nécessairement un individu de la population. Le processus est répété plusieurs fois jusqu'à stabilité des centres des classes (les centres ne bougent plus). En pratique, la méthode des centres mobiles cherche à minimiser *l'inertie intraclasse* définie par la somme des écarts des centroïdes aux points de leurs classes et donc à maximiser aussi *l'inertie interclasse* de la partition donnée par la somme des écarts entre les centroïdes des classes et le centroïde de la population totale (d'après le théorème de Huygens : *inertie totale* = *inertie intraclasse* + *inertie interclasse*). En minimisant *l'inertie intraclasse,* la méthode des centres mobiles a tendance à chercher des classes sphériques, d'égal volume et de faible inertie [Celeux *et al.*, 1989].

Cette méthode a connu des améliorations comme la *méthode des k-moyennes (k-means)* de Mac Queen [Hartigan and Wong, 1979; Mac-Queen, 1967]. Avec l'approche *k-means*, les centres sont recalculés après chaque affectation d'un individu dans une classe, plutôt que d'attendre l'affectation de tous les individus avant de mettre à jour les centres. Cette approche conduit généralement à de meilleurs résultats que la méthode des centres mobiles et la convergence est également plus rapide.

L'avantage de ces algorithmes est avant tout leur grande simplicité mais aussi leur complexité algorithmique qui reste raisonnable. Cependant ces méthodes souffrent de certains inconvénients : d'une part, le calcul de moyenne qu'elles utilisent est très sensible aux points aberrants et restreint leur application aux données numériques. D'autre part, la partition finale obtenue est très dépendante du choix des centres initiaux.

#### *2.4.2.2. Les méthodes des k-médianes*

Comme précisé auparavant la principale différence de ces méthodes avec les centres mobiles se situe au niveau du choix du représentant d'une classe. Les méthodes *k-médianes* présentent l'avantage d'être applicable à tout type de données et sont dans l'ensemble plus robustes aux points aberrants que les méthodes des *k-moyennes*, d'autant qu'elles recourent aux médianes (*medoïdes*) plutôt qu'aux moyennes (*centroïdes*) pour évaluer la distance aux centres.

L'algorithme PAM (Partition Around Medoids) est un exemple typique de ces méthodes dont le déroulement est le suivant [Kaufman and Rousseeuw, 1990] : après un choix aléatoire de $k$ médianes initiales, l'algorithme passe en revue tous les couples d'individus tels que l'un est une médiane et l'autre une non-médiane, en évaluant si l'échange des deux individus permet d'améliorer une *fonction objectif*. L'échange qui permet la plus grande amélioration de la fonction objectif est réalisé et une nouvelle itération a alors lieu. Au final, les différents objets sont affectés à la classe de médiane la plus proche. Cet algorithme présente l'avantage de pouvoir utiliser tout type de distance avec les *k-médianes*, puisqu'il n'est pas nécessaire de définir la moyenne des objets. Cependant, la complexité algorithmique est importante.

Afin de résoudre partiellement le problème des temps élevés de calcul avec les algorithmes de type PAM, l'algorithme CLARA (Custering LARge Applications), introduit par [Kaufman and Rousseeuw, 1990] effectue une recherche locale des médianes en opérant sur plusieurs échantillons de données de taille fixée. L'algorithme PAM est appliqué à chacun d'entre eux et le résultat retenu est le meilleur parmi les différents résultats. En conséquence, CLARA peut traiter des jeux de données de taille beaucoup plus importante que ceux traités par PAM. Les auteurs ont indiqué, suite à leurs expérimentations, que 5 échantillons de tailles respectives $40+2k$ individus suffisent pour donner un bon résultat. Le principal inconvénient de cet algorithme est que son efficacité dépend de la taille et de la forme des échantillons considérés.

Un autre algorithme dit CLARANS (Clustering Large Applications based upon RANdomized Search) proposé dans [NG and Han, 1994,2002] est aussi souvent utilisé comme étant une combinaison de PAM et CLARA. Cet algorithme utilise un graphe de classification pour représenter le problème de recherche des $k$ meilleurs médianes. Chaque sommet de ce graphe est représenté par un ensemble de $k$ individus (intuitivement, ce sont $k$ médianes). Deux sommets sont dits voisins si leurs ensembles ne

se diffèrent que par un seul individu. En conséquence, le problème de déterminer un ensemble de *k* meilleurs médianes devient le problème de recherche dans ce graphe du meilleur sommet. Le critère pour estimer qu'un sommet est meilleur qu'un autre est d'assurer une amélioration d'une *fonction objectif* (la somme des carrés des distances aux médianes) par l'échange d'une médiane par un individu *non-médiane* suite au déplacement d'un sommet à un autre voisin. Cependant, PAM le fait en parcourant tous les sommets voisins d'un sommet, ce qui devient très coûteux. CLARA opère sur une région localisée en parcourant un sous graphe, ce qui ne donne pas toujours le bon résultat. CLARANS est en fait une combinaison des deux algorithmes. Plutôt que d'essayer tous les échanges entre chaque médiane et chaque non-médiane, CLARANS choisit aléatoirement un sommet dans le graphe et n'examine pas tous les voisinages. Il détermine un nombre de voisins aléatoires (pas tous les voisins, il s'agit d'un paramètre de l'algorithme) pour rechercher un voisin dont le coût de remplacement de ce sommet par son voisin est minimisé. Si ce coût est négatif, (*i.e.* le remplacement ne peut plus optimiser le résultat), l'algorithme stipule que le sommet trouvé assure le meilleur résultat, et le processus peut s'arrêter. Les classes obtenues par CLARANS sont en général de meilleure qualité que celles obtenues avec PAM et CLARA. La limitation de cet algorithme est le nombre de paramètres à fixer a priori.

#### *2.4.2.3. Les nuées dynamiques*

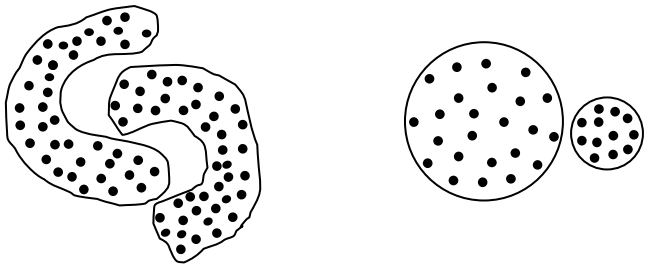
La méthode des *nuées dynamiques* largement développé par Diday dans [Diday, 1971] se distingue principalement des approches précédentes par le mode de représentation des classes appelé aussi *noyau*. Ce dernier peut être son centre de gravité (dans ce cas nous retrouvons l'approche des centres mobiles), un ensemble d'individus (l'approche des *k-médianes* avec un seul individu), une distance (l'approche des distances adaptatives [Diday and Govaert, 1977]), une loi de probabilité (la décomposition de mélanges [Schroeder, 1976]), etc.

L'algorithme des *nuées dynamiques* cherche à optimiser un critère objectif mesurant l'adéquation entre une partition et un mode de représentation des classes de cette partition. En pratique, l'algorithme converge lorsque ce critère à optimiser cesse de décroitre de façon sensible, ou lorsqu'un nombre fixé d'itérations est atteint. Celeux *et al.* précisent dans [Celeux *et al.*, 1989] que : « *le problème d'optimisation dans ce cas se pose en terme de recherche simultanée de la classification et de la représentation des classes de cette classification parmi un ensemble de classifications et de représentations possibles, qui minimisent le critère fixé* ». Afin de minimiser ce critère, l'algorithme des nuées dynamiques utilise principalement une étape de représentation suivie d'une étape d'affectation de façon itérative jusqu'à la convergence qui donne une solution localement optimale au problème posé.

La méthode des nuées dynamiques comme les approches précédentes de classification automatique par partitionnement fournit une solution dépendante de la configuration initiale qui est généralement faite par tirage au hasard.

#### *2.4.2.4. Conclusion*

On observe que dans la majorité des cas, les *k* classes trouvées par une approche de classification automatique par partitionnement sont de meilleure qualité que celles générées par une approche hiérarchique [NG and Han, 2002]. Cependant les algorithmes de classification par partitionnement souffrent du fait qu'ils n'utilisent dans l'ensemble qu'un seul point comme représentant d'une classe (problème de "représentant unique"). En effet, comme le but de ces algorithmes est de trouver les classes qui réduisent au minimum une fonction objectif égale à la somme des carrés des distances aux noyaux des classes, ils échouent pour les jeux de données où certains individus sont plus proches du noyau d'une autre classe qu'au noyau de leur propre classe. En conséquence, ils ne peuvent pas facilement capturer les classes avec des formes arbitraires (*cf.* figure 2.3 (a)) ou avec des tailles très différentes (*cf.* figure 2.3 (b)) [Karypis *et al.*, 1999].



(a) Classes de formes arbitraires (b) Classes de tailles différentes

**Figure 2.3 - Jeux de données pour lesquels les approches k-moyennes et k-médianes échouent**

### 2.4.3. Autres approches particulières de classification

Dans chacune des deux catégories d'approches de classification automatique que nous venons de détailler ci-dessus, les méthodes sont fondées sur la notion de *distance*. D'autres méthodes ont été développées et particulièrement introduites pour des problèmes et des bases de données particulières. Ces méthodes peuvent être divisées en deux groupes suivant la définition d'une classe. Nous trouverons ainsi (1) les méthodes fondées sur la notion de *densité* et (2) les méthodes fondées sur un *modèle*.

#### *2.4.3.1. Les approches fondées sur la notion de densité*

L'idée des méthodes par densité est de définir une classe comme étant un ensemble d'individus de forme quelconque, mais "dense" selon un critère de voisinage et de connectivité. Ces méthodes sont fondées sur les concepts de *densité*, *noyau*, *point limite*, *accessibilité* et *connectivité* détaillés dans [Han *et al.*, 2001].

Un des algorithmes les plus utilisés est DBSCAN (Density Based Spatial Clustering of Applications with Noise) [Ester *et al.*, 1996], et ses dérivés tels que DBCLASD [XU *et al.*, 1998] ou OPTICS [Ankerst *et al.*, 1999]. L'idée principale est de définir la notion de *voisinage* de rayon $\epsilon$ d'un point : tous les points situés à une distance de ce point inférieure à $\epsilon$ appartiennent alors à son voisinage. A partir de cette notion de $\epsilon - voisinage$, les auteurs définissent les concepts de *point central* (un point contenant au moins MinPts points dans son voisinage) et de *densité-accessible* entre deux points (s'il existe une chaîne de points centraux reliés par leur voisinage). Chaque classe est identifiée par des ensembles compacts de *points centraux* (appelés aussi *points noyaux*) ; les points situés à leur périphérie constituent sa bordure (appelés aussi *points limites*). L'algorithme a le déroulement suivant :

1. Choisir aléatoirement un point $x$ de l'ensemble d'individus $X$.
2. Rassembler dans un groupe $c$, tous les points *densité-accessible* à partir de $x$.
3. Si $x$ est un *point central*, alors $c$ est une classe,
4. *Sinon*, $x$ est un *point limite* alors aucun point n'est atteignable à partir de $x$. L'algorithme sélectionne un autre point $x \in X$ et reprend en 2 jusqu'à avoir balayer tous les points de $X$.

Ce type d'algorithme présente l'intérêt de trouver lui-même une évaluation du nombre de classes, et celles-ci peuvent avoir des formes arbitraires. Il permet également de gérer tout type de données et de bien tenir compte des données aberrantes, qui ne sont pas affectées aux clusters identifiés. Cependant comme pour BIRCH et CHAMELEON, l'algorithme requiert l'entrée des paramètres $\epsilon$ et MinPts, et l'expérience montre que les résultats obtenus sont très sensibles aux choix de ces paramètres [Nakache and Confais, 2005].

#### *2.4.3.2. Les approches fondées sur un modèle*

Les algorithmes des méthodes basées sur un modèle permettent de fournir de bonnes approximations des paramètres du modèle, afin que ce dernier s'ajuste au mieux aux données. Ils peuvent être de type partitionnement ou hiérarchique. Ils sont toutefois plus proches des algorithmes basées sur la densité, en ce sens qu'ils donnent naissance à des classes particulières qui améliorent un certain modèle préconçu. Dans ce cadre, nous pouvons distinguer deux familles d'approches : les approches neuronales (par exemple les cartes auto-organisatrices de Kohonen) et les approches probabilistes.

##### *a. Les approches neuronales*

L'algorithme des cartes topologiques proposé par Kohonen [Kohonen, 1997] est un procédé d'auto-organisation qui cherche à projeter des données multidimensionnelles sur un espace de faible dimension (en général de dimension 2), appelé *carte*. Cette réduction de dimension (ou aussi *projection*) permet d'obtenir un partitionnement des individus en groupes similaires, tout en préservant au mieux la structure topologique des données.

Dans cette topologie, les individus les plus semblables seront plus proches sur la carte et les données sont représentées par un nombre réduit de représentants appelés *neurones*, ce qui permet une meilleure visualisation des données.

L'algorithme d'apprentissage de la carte considère la carte comme une surface élastique qu'il étire pour la rapprocher des données. Il est de fait très proche des algorithmes des *centres mobiles*. Son déroulement est le suivant : après une initialisation aléatoire des valeurs de chaque neurone, les données sont soumises l'une après l'autre à la carte auto-adaptative. Selon les valeurs des représentants, il en existe un qui est le plus proche de la donnée présentée. Ce neurone est dit *gagnant* et sera alors tiré avec ses neurones voisins vers cette donnée afin qu'ils répondent encore mieux à d'autres données de même nature : ceci garantit de préserver la topologie. En fin d'apprentissage, lorsque les neurones ne bougent que légèrement entre les itérations, ce réseau compétitif garantit de reproduire sur la carte de sortie les corrélations présentes dans les données d'entrées.

Après la classification de Kohonen, il est possible d'effectuer une classification hiérarchique sur les neurones finaux des classes, de manière à les regrouper en classes moins nombreuses.

Dans [El-Golli, 2004; El-Golli *et al.*, 2004], l'auteur a proposé une adaptation des cartes topologiques de Kohonen aux données complexes. En effet, les SOM (Self Organized Map) se basent sur la notion de *centre de gravité* qui est difficile à définir quand on est en présence de descriptions symboliques. L'auteur a ainsi proposé une modification de l'algorithme des cartes auto-organisatrices afin de permettre son application aux tableaux de dissimilarités.

### *b. Le mélange de distributions*

Dans les approches probabilistes, les données sont considérées comme étant un échantillon indépendamment tiré d'un modèle de mélange de plusieurs distributions de probabilités.

Elles reposent sur l'hypothèse que les données de chaque classe suivent une distribution de probabilité et que l'ensemble total des données forme un mélange de distributions. En d'autres termes, elles considèrent que les individus sont générées par la sélection au hasard d'un modèle $R$ $\{R=1...k\}$ ($k$ étant le nombre de modèles ou classes qui est fixé a priori) avec une probabilité $P_R$ et essayent d'estimer les paramètres des différentes distributions du mélange (les probabilités a priori) par l'algorithme d'estimation EM (*Expectation-Maximisation*) [Dempster *et al.*, 1977] ou par un algorithme de type nuées dynamiques [Celeux *et al.*, 1989; Schroeder, 1976]. Un avantage clair de telles approches est la facilité d'*interprétation* des classes construites. Bien sûr, la convergence de ces approches peut être très lente et les résultats obtenus sont sensibles aux choix des probabilités initiales [Fraley and Raftery, 1998]. Par ailleurs, le nombre de paramètres à estimer augmente linéairement avec la taille de l'ensemble des données.

## 2.4.4. Les approches fondées sur la théorie des graphes

Lorsqu'un tableau de ressemblances (en général similarité/dissimilarité) $D = \{d(X_i, X_j)\}$ est défini sur l'ensemble des individus $X = \{X_1, X_2, \dots, X_n\}$, les données peuvent être également conçues comme un graphe valué $G(\boldsymbol{V}, \boldsymbol{E})$ où $V = \{v_1, v_2, \dots, v_n\}$ est l'ensemble des sommets qui sont les individus à analyser ($v_i$ pour $X_i$) et $E = VxV$ est l'ensemble d'arêtes qui correspond, quant à lui, aux paires de sommets $(v_i, v_j)$ pondérés par la ressemblance $D(X_i, X_j)$. Dans ce cadre, d'autres mécanismes de classification automatique basés sur la théorie des graphes ont été proposées dans la littérature [Auguston and Minker, 1970]. Ils consistent principalement à chercher des structures combinatoires dans les graphes de similarité/dissimilarité, tel que : *l'arbre couvrant* [Zahn, 1971], les *composantes connexes* [Matula, 1970,1972], la *coupe minimale* [Shi and Malik, 2000; Wu and Leahy, 1993], ou les *sous-graphes complets* (appelés aussi *cliques*) [Gotlieb and Kumar, 1968]. Nous proposons dans cette section d'étudier plus finement les mécanismes des approches de classification exploitant la théorie des graphes.

### *2.4.4.1. Définitions*

Rappelons d'abord certaines notions sur les graphes qui seront utilisées dans ces approches.

***Définition 2.3*** *Un arbre couvrant du graphe G est un sous-graphe connexe et sans cycle de G. Un arbre couvrant est dit minimal si la somme des longueurs de ses arêtes est minimale. Si cette somme est maximale, nous parlons d'arbre couvrant maximal.*

***Définition 2.4*** *Une composante connexe H du graphe G est un sous-graphe induit $H \subseteq G$ tel que ses sommets sont accessibles les uns depuis les autres. Autrement dit, pour toute paire de sommets (a,b) de H, il existe une suite d'arêtes entre les sommets a et b.*

***Définition 2.5*** *Un sous-graphe complet de G appelé aussi clique est un sous-graphe dont les sommets sont deux-à-deux adjacentes (reliés par une seule arrête). Une clique est dite maximale dans G si elle n'est pas contenue dans un autre sous-graphe complet de G.*

***Définition 2.6*** *La notion de diamètre d'une classe a été largement utilisée dans le cadre des approches de classification automatique à base de graphes. Pour un graphe où les arêtes reflètent la dissemblance entre les sommets (i.e. il existe une arête entre une paire de sommets s'ils sont* dissimilaires *au sens d'un seuil fixé), la définition du diamètre d'une classe est liée, dans ce cas, à la notion de distance. Il est donné par la plus grande dissimilarité intraclasse ; le diamètre d'une partition est alors égal au plus grand des diamètres de ses classes. Pour un graphe où les arêtes reflètent la ressemblance entre les sommets (i.e. il existe une arête entre une paire de sommets s'ils sont* similaires *au sens d'un seuil fixé), la définition du diamètre d'une classe est associée à la notion de connectivité. Ainsi, le diamètre*

*d'une classe C est donné par la plus grande distance entre deux sommets dans C, sachant que la distance entre deux sommets est la longueur minimale d'un chemin qui les relie.*

#### *2.4.4.2. Quelques algorithmes de clustering à base de graphes*

Un algorithme divisif très connu dans le domaine de classification automatique à base de graphes consiste à construire dans une première étape, *l'arbre couvrant minimal* à partir du *tableau de dissimilarités*, puis de supprimer, à chaque itération, l'arête de l'arbre de longueur maximale afin de générer une hiérarchie de partitions (*dendrogramme*). Certains auteurs ont étudié aussi l'utilisation de la coloration des sommets d'un arbre couvrant dans le cadre des approches hiérarchiques de classification. Dans ce cadre, Guénoche *et al.* ont montré dans [Guénoche *et al.*, 1991] que la partition en deux classes obtenue par *bi-coloration* d'un *arbre couvrant maximal* a un diamètre minimum. Ils ont développé pour cela une stratégie divisive de partitionnement qui consiste à subdiviser à chaque itération la classe de diamètre maximum en deux classes afin de trouver une partition de diamètre minimum. La subdivision d'une classe est obtenue à l'aide d'une procédure de *bi-coloration* des sommets de l'arbre couvrant maximum construit par *l'algorithme de Prim* (1957) à partir du tableau de dissimilarités. Elle consiste à colorer tous les sommets de l'arbre couvrant maximum avec deux couleurs telles que deux sommets adjacents ne portent pas la même couleur.

Une première proposition d'utilisation des composants connexes pour découvrir les clusters a été faite par D.W. Matula [Matula, 1970,1972] dans les années 70. Il propose d'exploiter le tableau de similarités entre les individus pour construire un *graphe seuil supérieur* dont les sommets correspondent aux individus, les arêtes reliant les individus ayant des valeurs de similarité supérieures à un seuil $\theta$ donné. Matula a observé comment l'analyse de connectivité dans ce graphe seuil de similarité pourrait être utile dans un problème de classification automatique. Se basant sur la notion d'*arête-connectivité* d'un graphe -définie par le nombre minimum d'arêtes dont la suppression déconnecte le graphe-, et de la fonction de *cohésion* -définie pour chaque sommet et arête d'un graphe $G$ comme le maximum des *arête-connectivités* des sous-graphes contenant cet élément-, Matula [Matula, 1970] a proposé d'identifier les clusters par les *sous-graphes k-connexes maximaux* de $G$, obtenus par la suppression de tous les éléments dans $G$ ayant une cohésion inférieure à $k$ (où $k$ est un entier positif défini à l'avance). Le problème principal de cette approche est que les « *vrais* » clusters des données peuvent avoir différentes valeurs de *connectivité*. Comme solution à ce problème, Matula a proposé dans [Matula, 1972] de définir les clusters par les *sous-graphes k-connexes maximaux* (pour tout entier $k$) ne contenant aucune *sous-composante* à forte connectivité. Même si cette solution répond au problème précédent, elle présente alors le risque de diviser certains "vrais" clusters ayant plusieurs parties fortement cohésives [Hartuv and Shamir, 2000].

Dans le même ordre d'idée, Hartuv et Shamir [Hartuv and Shamir, 2000] ont développé un algorithme de clustering, qu'ils appellent HCS (*Highly Connected Subgraphs*), se basant

sur la notion de *composante connexe*. Dans le graphe seuil de similarité, ils identifient les clusters par les sous-graphes fortement connexes dont *l'arête-connectivité* excède la moitié du nombre de sommets. De tels sous-graphes sont déterminés en utilisant des algorithmes de recherche des ensembles minimaux d'arêtes dont la suppression déconnecte le graphe initial. Les auteurs ont montré que la partition fournie par leur approche présente plusieurs propriétés traditionnellement recherchées dans un problème de classification automatique : les classes produites par HCS ont un diamètre maximal de *deux*, ce qui reflète leur forte homogénéité vu que deux sommets quelconques sont soit adjacents, ou bien ils ont en commun un ou plusieurs voisin(s). Les auteurs ont également prouvé que cette propriété n'est pas vérifiée par les algorithmes de clustering de Matula et de Wu & Leahy présenté ci-dessous [Wu and Leahy, 1993].

D'autre part, Wu et Leahy [Wu and Leahy, 1993], ont proposé d'utiliser la notion de *coupe minimale* dans les *graphes de similarité pondérés*[2] pour le problème de classification automatique. Dans un *graphe de similarité pondéré* $G$, une coupe est l'ensemble d'arêtes qui partitionnent l'ensemble des sommets $V$ de $G$ en deux parties $V_1$ et $V_2$ tel que $V_1 \cup V_2 = V$ et $V_1 \cap V_2 = \emptyset$. Le *coût* (ou *capacité*) d'une coupe est donnée par la somme des poids (similarités) des *arrêtes de la coupe* (*i.e.* les arêtes reliant les deux parties $V_1$ et $V_2$ $\{(u,v)|u \in V_1 ; u \in V_2\}$). Pour produire une partition en $k$ classes des données ($k$ étant défini à l'avance), Wu et Leahy proposent de calculer les ($k$-*1*) coupes minimales (qui minimisent le coût de séparation) dans $G$ par le biais d'un algorithme présenté dans [Gomory and Hu, 1961]. Cet algorithme a été utilisé pour obtenir de bons résultats de segmentation sur les images. Cependant, étant donné que le coût d'une coupe est toujours fonction de nombre de ses arêtes, ce concept de coupe minimale a tendance à partitionner le graphe de telle sorte que le plus petit nombre d'arcs possible interviennent dans la coupe. Ceci génère souvent des classes avec peu de points et conduit à un partitionnement déséquilibré [Shi and Malik, 2000]. Pour remédier à ce problème, Shi et Malik [Shi and Malik, 2000] ont alors proposé une normalisation du coût de la coupe minimale, qui supprime l'influence du nombre d'arêtes.

Les approches de clustering à base de graphes les plus répandues aujourd'hui consistent à chercher les *sous-graphes complets* d'un graphe seuil donné. Puisque les *sous-graphes complets* (*cliques*) sont reliés à la notion de *composantes fortement connexes*, de telles structures sont généralement considérées pour fournir des classes fortement homogènes. Plusieurs travaux explorant cette idée sont rapportés dans la littérature. Dans [Kuhns, 1959], Kuhns était le premier à définir une clique maximale d'un graphe donné comme un cluster. Cependant, il n'a pas fourni des résultats expérimentaux. Gotlieb et Kumar [Gotlieb and Kumar, 1968] ont exploité également le concept des *cliques maximales* pour définir des clusters à partir d'un graphe non pondéré résultant d'une opération de seuillage sur le graphe de similarité initial. La recherche de cliques maximales a été

[2] Les graphes de similarité pondérés sont des graphes complets dont les arêtes sont les liens pondérés par les similarités entre les paires de données.

également liée au problème de *coloration des sommets de graphes*. En effet, Hansen et Delattre [Hansen and Delattre, 1978] ont ramené la recherche d'une partition de diamètre minimum à celui d'une *coloration minimale* d'un graphe seuil supérieur ayant pour sommets l'ensemble d'individus de la population, et pour arêtes les paires de sommets dont la dissimilarité est supérieure à un seuil $\theta$ donné. Cette coloration affecte des couleurs différentes aux sommets adjacents avec un nombre minimum de couleurs utilisées (chaque couleur correspondra à une classe de la partition obtenue). Il est à noter qu'alors que les méthodes basées sur la recherche de cliques tendent à construire une partition des données en classes homogènes, elles n'accordent aucune importance à la *séparation intercluster*.

## 2.5. Evaluation des approches de classification automatique

L'objectif des approches de classification automatique est de produire des classes avec une *cohésion* maximale (*similarité intraclasse*) tout en réalisant un maximum de *séparation* (*dissimilarité interclasse*) entre les classes de la partition obtenue.

Dans les techniques de classification hiérarchique, nous savons qu'une coupure du dendrogramme par une droite horizontale fournit une partition de l'ensemble des individus à classer. Le nombre de classes de la partition est défini par le niveau de cette coupure, qui n'est pas toujours facile à déterminer. Souvent plusieurs niveaux de coupure sont retenus et les partitions qui s'en déduisent sont comparées afin de retenir la meilleure en termes de *compacité* et de *séparabilité* des classes. Pour les techniques de classification par partitionnement, le nombre de classes à découvrir doit être fixé a priori, ce qui n'est pas toujours le cas des ensembles de données. C'est pourquoi généralement, on fixe plusieurs valeurs pour le nombre $k$ de classes, et les partitions correspondantes sont ensuite comparées.

On constate ainsi qu'un certain nombre d'approches nécessite l'évaluation de la qualité des partitions de données obtenues. Or l'analyse et la comparaison des partitions n'est pas un processus immédiat. Tout comme il existe de nombreux algorithmes de classification automatique, la littérature fourmille de critères associés à la qualité du clustering. Nous ne nous intéresserons ici qu'aux critères dits *internes*, c'est-à-dire basés sur les données et les ressemblances entre elles (similarités/dissimilarités). Les critères *externes* quant à eux sont basés sur des informations extérieures comme le *label de classes* ou l'*avis d'un expert*. Lorsque nous utilisons les critères internes, le problème de classification automatique est alors considéré comme un problème *d'optimisation* dont les performances peuvent être déterminées. De nombreuses procédures, appelées aussi *indices de validité de clustering*, ont été proposées dans la littérature dans le but de déterminer la meilleure partition d'un jeu de données numériques [Bezdek and Pal, 1998]. Ces procédures sont bien adaptées au cas symbolique par le fait qu'elles ne nécessitent que la définition d'une mesure de dissimilarité pour fonctionner [Mali and Mitra, 2003].

Pour la présentation de ces différents indices, nous proposons d'expliciter quelques notations préliminaires. Pour une partition $P$ en $k$ classes $\{C_1, C_2, \dots, C_k\}$ de l'ensemble d'individus $X = \{X_1, X_2, \dots, X_n\}$, la *dispersion intraclasse* $s_a(C_i)$ et la *séparation interclasse* $d_a(C_i, C_j)$ sont données par les formules suivantes :

$$\forall\ C_i \in P;\quad s_a(C_i) = \frac{1}{|C_i|(|C_i|-1)} \sum_{u=1}^{|C_i|} \sum_{q=1}^{|C_i|} d(X_u, X_q) \tag{2.30}$$

$$\forall\ C_i, C_j \in P;\quad d_a(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{u=1}^{|C_i|} \sum_{q=1}^{|C_j|} d(X_u, X_q) \tag{2.31}$$

où $d$ représente la mesure de dissimilarité définie sur l'ensemble d'individus $X$ et $|C_i|$ le cardinal (nombre d'individus) de la classe $C_i$.

### 2.5.1. L'indice de Dunn

L'indice de Dunn [Dunn, 1973] est basé sur l'identification de clusters compacts et bien séparés. Il est défini par le rapport entre la plus petite *dissimilarité interclasse* $d_{min}$ (*i.e.* entre deux individus de deux classes différentes) et la plus grande *dissimilarité intraclasse* $s_{max}$ (*i.e.* entre deux individus de la même classe).

$$Dunn(P) = \frac{d_{min}}{s_{max}} \tag{2.32}$$

L'objectif principal de cet indice est de maximiser la *dissimilarité interclasse* et de minimiser la *dissimilarité intraclasse*. L'objectif est donc de maximiser l'indice de *Dunn*.

### 2.5.2. L'indice de Dunn généralisé

L'indice de *Dunn généralisé* [Bezdek and Pal, 1998] a été introduit après qu'il ait été démontré que l'indice de *Dunn* [Dunn, 1973] était non robuste : en effet il dépend uniquement d'un nombre réduit d'individus de la population, et des liens établis entre eux. Ainsi, il est sensible à tout changement qui intervient dans la structure des clusters ainsi qu'aux points aberrants. Les modifications apportées à cet indice interviennent dans le calcul de la dissimilarité *interclasse* et *intraclasse*.

L'indice de *Dunn généralisé* est reconnu comme l'un des indices les plus appropriés pour l'évaluation de la qualité d'une partition donnée, car il fournit un bon compromis entre la *maximisation* de la *dissimilarité interclasse* et la *minimisation* de la *dissimilarité intraclasse* de la partition.

$$Dunn\_gen(P) = \frac{\min_i \left\{ \min_{j \neq i} \{ d_a(C_i, C_j) \} \right\}}{\max_h s_a(C_h)} \tag{2.33}$$

La partition $P$ produisant la plus grande valeur de $Dunn\_gen(P)$ correspondra à la meilleure classification.

### 2.5.3. L'indice de Davies-Bouldin

L'indice de *Davies-Bouldin* [Bezdek and Pal, 1998] est basé sur la minimisation du rapport des *dispersions intraclasse* et de la *séparation interclasse*. Il est calculé comme suit:

$$DB(P) = \frac{1}{k} \sum_{i=1}^{k} \max_{\substack{1 \leq j \leq k \\ j \neq i}} \left\{ \frac{s_a(C_i) + s_a(C_j)}{d_a(C_i, C_j)} \right\} \tag{2.34}$$

On constate ainsi que le ratio sera d'autant plus faible que les classes seront compactes et éloignées les unes des autres. Par conséquence, la partition de meilleure qualité sera celle qui minimisera l'indice de *Davies-Bouldin*.

### 2.5.4. L'indice de Silhouette

L'indice de silhouette est défini par [Rousseeuw, 1987] pour tout individu $X_i$ de l'ensemble $X$ par la formule suivante :

$$\forall\ X_i \in X;\ \ s(X_i) = \frac{b(X_i) - a(X_i)}{\max\left(a(X_i), b(X_i)\right)} \tag{2.35}$$

où

- $a(X_i)$ est la dissimilarité moyenne entre l'individu $X_i$ et tous les autres individus de la classe à la quelle il appartient $C(X_i)$.

$$\forall\ X_i \in X;\ \ a(X_i) = \frac{1}{|C(X_i)| - 1} \sum_{\substack{X_j \in C(X_i) \\ X_j \neq X_i}} d(X_i, X_j) \tag{2.36}$$

- $b(X_i)$ est le minimum des dissimilarités moyennes entre l'individu $X_i$ et tous les autres individus des classes de la partition $P$ différente de $C(X_i)$.

$$\forall\ X_i \in X;\ \ b(X_i) = \min_{\substack{C \in P \\ C \neq C(X_i)}} d(X_i, C)$$

$$o\grave{u}\ d(X_i, C) = \frac{1}{|C|} \sum_{X_j \in C} d(X_i, X_j) \tag{2.37}$$

On notera que l'indice de silhouette est borné : $-1 \leq s(X_i) \leq 1$. De plus, lorsque $s(X_i)$ est proche de 1, $X_i$ est dit *bien classé* dans $C(X_i)$. Quant $s(X_i)$ est proche de 0, alors $X_i$ se

situe entre deux classes. Finalement, si $s(X_i)$ est proche de -1, $X_i$ est dit *mal classé* dans $C(X_i)$ et doit être rattaché à un autre cluster le plus proche.

Chaque classe est aussi représentée par une silhouette qui montre quels objets sont correctement classés à l'intérieur de cette classe et lesquels n'ont simplement qu'une position intermédiaire. Pour une classe $C_i$ donnée, son indice de silhouette est défini par la moyenne des indices de silhouette des individus qui lui appartiennent :

$$\forall\ C_i \in P;\ s(C_i) = \frac{\sum_{X_j \in C_i} s(X_j)}{|C_i|} \tag{2.38}$$

L'indice de silhouette global de la partition $P$ est donné par la moyenne globale des largeurs de silhouettes dans les différentes classes $C_i$ qui composent la partition :

$$s(P) = \frac{\sum_{C_i \in P} s(C_i)}{k} \tag{2.39}$$

La meilleure partition retenue est alors celle qui permet d'obtenir une silhouette globale maximale.

## 2.6. Conclusion

Ce chapitre était consacré à l'état de l'art des méthodes de classification automatique, selon le type des données à regrouper et la démarche choisie. Nous avons ainsi présenté le panel des approches hiérarchiques, puis celui des approches par partitionnement. Nous avons aussi donné un aperçu de nouvelles méthodes évoluées de classification, basées sur la notion de densité ou les modèles probabilistes. Pour chacune nous avons tenté de montrer les points forts et les points faibles.

Nous avons également présenté les méthodes de classification exploitant les techniques de la théorie des graphes, qui nous intéressent plus particulièrement.

Nous avons ainsi introduit l'ensemble des notions requises pour la suite de ce mémoire, et notamment les notions utilisées pour les graphes et les différents indices qui permettent d'évaluer la qualité d'une partition des données, ce qui est nécessaire dans le cas d'une approche itérative.

La suite de la présentation va exposer la démarche que nous avons élaborée pour la classification automatique de données, et présenter les applications effectuées sur différents jeux de données.

# CLASSIFICATION AUTOMATIQUE PAR B-COLORATION DE GRAPHES

## Résumé

*Nous présentons dans ce chapitre une nouvelle approche de classification automatique sur tableau de dissimilarités basée une technique de coloration de graphes, baptisée la b-coloration. Nous illustrons par quelques applications l'intérêt de l'approche développée tant dans la classification des données classiques que la classification des données complexes ou symboliques issues du système d'information hospitalier français.*

*Nous décrivons également dans ce chapitre l'extension associée à la méthode et qui concerne l'apprentissage automatique. Celle-ci permet la mise à jour incrémentale de la partition obtenue par l'approche sans avoir à relancer la classification sur toutes les données. L'orientation d'un nouvel individu est automatiquement effectuée vers la classe adéquate si elle est trouvée, sinon il y a un réarrangement de la partition initiale, ainsi que dans le cas où des données existantes sont supprimées.*

## Sommaire

# Chapitre 3

# Classification automatique par b-coloration de graphes

*" Il n'y a pas de conditions, de classes, de rang, dans la nature. Les hommes seuls ont cherché à mettre de l'ordre, là où il y en avait déjà et ils ont établi le désordre ! "*

Gilbert Louvain, "La Catherine de Montréal"

## 3.1. Introduction

A la lumière de l'état de l'art présenté au chapitre précédent, il est de situer notre approche par rapport aux familles de méthodes présentées et donc notre apport. Ainsi dans la suite, nous nous plaçons dans le cadre des méthodes de classification automatique à base de graphes, notre approche est itérative et va donc exploiter certains indices de qualité de partition présentés au chapitre précédent pour déterminer quand arrêter le processus. D'autre part la méthodologie de classification proposée répond aux hypothèses suivantes :

- une recherche automatique du nombre de classes.
- assurer une forte *cohésion intraclasse* et une nette *séparation interclasse*.
- traiter aussi bien les données de *descriptions classiques* (numériques et catégorielles) que les données de *descriptions symboliques* (ensembles de valeurs).
- fournir une représentation des classes de la partition obtenue.
- tenir compte de l'aspect incrémental de classification : cette notion est primordiale pour les applications qui ont un flux données entrant continu, comme les systèmes d'analyse de documents ou l'analyse en ligne (banque, finance, télécommunication, etc.), mais aussi quand nous traitons des jeux de données de grande taille où les contraintes en termes de temps d'exécution ou d'espace mémoire sont importantes.

Ces hypothèses nous ont amenés à considérer de plus près les méthodes de classification à base de graphes, qui nous ont semblé ouvrir d'intéressantes opportunités

pour la problématique de classification sans connaissance du nombre de classes a priori. Nous avons donc décidé de nous associer au défi scientifique qui consiste à mixer les structures de *sous-graphes complets* et le problème de classification automatique. C'est ainsi que nous nous sommes penchés sur une technique récente de coloration de graphes, appelée *b-coloration* [Elghazel *et al.*, 2006b; Elghazel *et al.*, 2006c].

Cette technique de coloration possède l'avantage de fournir une partition fine des données où la *séparation interclasse* est réalisée simultanément avec la *cohésion intraclasse*, quand le nombre de classes n'est pas fixé a priori. Elle possède également un ensemble de caractéristiques particulièrement intéressantes, à savoir :

- la méthode de *b-coloration* évince le problème du "*représentant unique*" des classes [Karypis *et al.*, 1999] mentionné dans le paragraphe 2.4.2.3. En effet, l'algorithme de coloration construit les classes de la partition en se basant sur les relations topologiques entre les individus pris deux à deux (*i.e.* les dissimilarités) et non pas sur la distance entre les individus et un représentant unique de la classe ;
- elle est adaptée à tout type de données, *classiques* et *symboliques*, dès lors qu'un tableau de dissimilarités peut être construit sur les données ;
- elle permet également de marquer chaque classe par au moins un sommet représentant –que nous appelons *sommet dominant*. Ce sommet est le reflet des propriétés de la classe et en fournit donc un représentant qui facilite l'interprétabilité des classes, et garantit d'autre part une séparation nette de la classe vis-à-vis des autres classes de la partition obtenue.

Nous proposons également dans ce chapitre une extension de la méthode de classification qui concerne *l'apprentissage incrémental* [Elghazel *et al.*, 2007c]. En effet, le point fort des algorithmes fondés sur la théorie des graphes est leur capacité d'être étendus à une version dite *en ligne,* afin de traiter des nouvelles données insérées, ou les données retirées de la population initiale (données obsolètes ou inefficaces). Ceci consiste à mettre à jour les classes déjà obtenues sans avoir à relancer le processus global de classification. L'idée fondamentale est la suivante : une fois la partition optimale retournée par l'algorithme de *b-coloration de graphes*, nous travaillons à assigner de nouvelles données à leurs classes (couleurs) adéquates au moment où elles arrivent au système ou de réarranger la partition quand des données existantes sont retirées de la population. Ceci est réalisé avec un temps d'exécution minimisé tout en maintenant les propriétés de la *b-coloration* et en respectant les contraintes de *qualité de classification* mentionnées précédemment.

Le reste de ce chapitre est structuré comme suit : dans une première partie, nous présentons d'abord la technique de *b-coloration de graphe* (&3.2). Nous exposons par la suite l'approche de classification automatique proposée en détaillant les différentes phases du processus de b-coloration itératif ainsi que le critère d'arrêt (&3.3). La section

suivante est consacrée aux expérimentations de notre approche menées sur des jeux de *données benchmark* (utilisés pour évaluer les approches de classification automatiques), sur des jeux de *données médicales* extraits du système de santé français ainsi que sur des images annotées issues du domaine archéologique (&3.4). Nous verrons ainsi que la méthode donne de bons résultats comparé aux autres approches de classification, et qu'elle répond aux objectifs fixés en donnant notamment des *classes homogènes* et *bien séparées*. Enfin dans la dernière partie de ce chapitre, nous détaillons l'algorithme d'apprentissage incrémental (&3.5).

## 3.2. La b-coloration de graphes

Soit $G = (\boldsymbol{V}, \boldsymbol{E})$ un graphe simple non orienté, où $\boldsymbol{V}$ est l'ensemble des sommets et $\boldsymbol{E}$ l'ensemble des arêtes. Une *b-coloration* de $G$ est une fonction de coloration $C: V \rightarrow Z^+$ qui consiste à colorer tous les sommets de $V$ à l'aide d'une coloration maximum de telle sorte que :

- pour toute paire de sommets adjacents $(v_i, v_j) \in E, C(v_i) \neq C(v_j)$ (*coloration propre*).
- pour toute classe de couleur $C$, il existe un sommet $S \in V$, appelé *sommet dominant*, coloré par cette couleur et adjacent à toutes les autres couleurs. Une couleur avec un sommet dominant est dite *couleur dominante*.

Nous appelons le *nombre b-chromatique* $\varphi(G)$ le nombre maximum $k$ tel que $G$ possède une *b-coloration* avec $k$ couleurs.

Contrairement à la *coloration minimale* de sommets d'un graphe $G$, la *b-coloration* consiste à colorer les sommets du graphe avec un maximum de couleurs sous les contraintes de *propretés* et de *dominance*.

Sachant que le degré d'un sommet est égale au nombre de ses voisins et le que degré maximum du graphe $G$, notés par $\Delta(G)$, est le plus grand des degrés de ses sommets, Irving et Manlove [Irving and Manlove, 1999] ont introduit en 1999 le concept de *b-coloration* et ont montré les propriétés suivantes :

***Proposition 3.1*** *Soit $G$ un graphe et $\chi(G)$ son nombre chromatique, défini comme le nombre minimum de couleurs requises pour une coloration propre de $G$. Alors nous avons :*

$$\chi(G) \leq \varphi(G) \leq \Delta(G) + 1 \qquad (3.1)$$

***Preuve*** *Comme la b-coloration de $G$ est une coloration propre : $\chi(G) \leq \varphi(G)$. Il est facile de voir que pour tout sommet de degré $\Delta(G)$ peut avoir au maximum $\Delta(G)$ couleurs voisines et prendre pour lui la couleur $\Delta(G) + 1$. En conséquence, $\varphi(G) \leq \Delta(G) + 1$.*

Irving et Manlove ont également montré que la recherche du nombre *b-chromatique* $\varphi(G)$ pour tout graphe $G$ est un problème *NP-difficile*. Plusieurs auteurs ont étudié ce

paramètre pour des classes particulières des graphes comme les *arbres* [Irving and Manlove, 1999], les *graphes puissances* [Effantin and Kheddouci, 2003] et les *produits cartésiens de graphes* [Kouider and Maheo, 2002]. Plus récemment, Effantin et Kheddouci [Effantin and Kheddouci, 2006] ont proposé un algorithme distribué pour construire une *b-coloration* d'un graphe $G$ quelconque. Pour plus de détails, dans [Kheddouci, 2003], l'auteur propose un large aperçu sur ce paramètre.

Si $G$ est un graphe *non connexe* composé de $p$ composantes connexes $(C_1, C_2, \dots, C_p)$, nous avons [Kouider and Maheo, 2002] :

$$\varphi(G) \leq \max_{i \leq l \leq p} \varphi(C_i) \tag{3.2}$$

## 3.3. Présentation de la méthode

Dans cette section, une nouvelle méthode de classification automatique par *b-coloration* de graphes est proposée [Elghazel *et al.*, 2006b; Elghazel *et al.*, 2006c].

Considérons la représentation topologique de l'ensemble d'individus à grouper $X = \{X_1, X_2, \dots, X_n\}$ par un graphe *complet*, *non orienté* et *pondéré* $G = (\boldsymbol{V}, \boldsymbol{E})$ pour lequel les sommets $\{v_1, v_2, \dots, v_n\}$ sont les individus à classer (le sommet $v_i$ correspond à l'individu $X_i$) et les arêtes les liens pondérés par les dissimilarités entre les paires de données. Le graphe $G$ est traditionnellement représenté par un tableau de dissimilarités symétrique de taille $nxn$ $D = \{d(X_i, X_j) | X_i, X_j \in X\}$.

Une définition simple et claire suppose que "un cluster est un ensemble d'éléments semblables, et les éléments de différents clusters sont différents". En conséquence, un cluster devrait satisfaire deux conditions fondamentales : (1) il devrait avoir une homogénéité interne élevée ; (2) il devrait avoir une hétérogénéité forte entre les éléments de différents clusters. Ces deux conditions s'élèvent à affirmer que les arêtes entre deux sommets d'un même cluster devraient avoir des faibles pondérations (indiquant une similarité élevée), et ceux entre les sommets de deux clusters devraient être à forte pondération (indiquant une similarité faible). Le problème de clustering est par conséquent ramené à un problème de b-coloration de graphes.

### 3.3.1. Construction d'un graphe seuil

La représentation topologique de l'ensemble d'individus à grouper $X = \{X_1, X_2, \dots, X_n\}$ par un graphe complet ne convient pas au problème de classification non supervisée. En effet, la *b-coloration* d'un tel graphe retournerait la classification "*triviale*" où chaque classe (couleur) est supposé contenir un seul individu (singleton). Notre algorithme de classification passe donc par la construction d'un *graphe seuil supérieur* défini comme le graphe partiel du graphe de départ $G = (\boldsymbol{V}, \boldsymbol{E})$. Un graphe seuil supérieur $G_{>\theta} = (\boldsymbol{V}, \boldsymbol{E}_{>\theta})$

est un graphe simple ayant pour ensemble de sommets celles du graphe d'origine (*i.e.* $V = \{v_1, v_2, \dots, v_n\}$) et pour ensemble d'arêtes $E_{>\theta}$ les paires de sommets dont la dissimilarité est supérieure à un seuil $\theta$ choisi à partir de la table de dissimilarités entre individus (*i.e.* $\forall\, v_i, v_j \in V$, l'arête $(v_i, v_j) \in E_{>\theta}$ *si et seulement si* $d(X_i, X_j) > \theta$ où $d(X_i, X_j)$ est la dissimilarité entre les deux individus $X_i$ et $X_j$ correspondant respectivement aux sommets $v_i$ et $v_j$).

Dans la suite, nous noterons que deux sommets sont *adjacents* ou *voisins* (*resp. non adjacents* ou *non voisins*) s'ils ont un degré de dissimilarité *supérieur* (*resp.* "*inférieur ou égal*") au seuil $\theta$.

A titre d'illustration, la figure 3.1 montre le graphe seuil supérieur $G_{>0.15}$ ($\theta = 0.15$) associé à la matrice de dissimilarités du tableau 3.1 suivant :

| $v_i$ | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| **A** | 0 | | | | | | | | |
| **B** | 0.20 | 0 | | | | | | | |
| **C** | 0.10 | 0.30 | 0 | | | | | | |
| **D** | 0.10 | 0.20 | 0.25 | 0 | | | | | |
| **E** | 0.20 | 0.20 | 0.10 | 0.40 | 0 | | | | |
| **F** | 0.20 | 0.20 | 0.20 | 0.25 | 0.65 | 0 | | | |
| **G** | 0.15 | 0.10 | 0.15 | 0.10 | 0.10 | 0.75 | 0 | | |
| **H** | 0.10 | 0.20 | 0.10 | 0.10 | 0.05 | 0.05 | 0.05 | 0 | |
| **I** | 0.40 | 0.075 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0 |

**Tableau 3.1 - Tableau de dissimilarités**

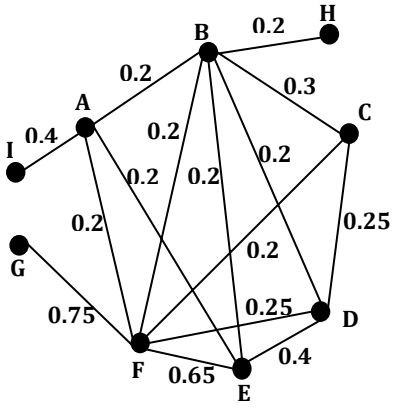



**Figure 3.1 - Graphe seuil supérieur $G_{>0.15}$ ($\theta$ =0.15)**

## 3.3.2. L'algorithme de b-coloration

Un nouvel algorithme de *b-coloration*, bien adapté au problème de clustering, est proposé dans cette section. L'idée principale est d'appliquer cet algorithme sur le graphe seuil supérieur $G_{>\theta}$. En effet, le but est de fournir une attribution de couleurs (classes) aux sommets de $G_{>\theta}$ de telle sorte que deux sommets adjacents (paire de sommets dont la

dissimilarité est supérieure au seuil $\theta$) n'aient pas la même couleur, et que pour chaque classe de couleur, il existe au moins un *sommet dominant* adjacent à au moins un sommet dans chacune des autres couleurs. Une partition associée au seuil $\theta$ est alors retournée, avec de bonnes propriétés de classification, notamment *l'homogénéité intracluster* et la *séparation intercluster*, comme nous allons le montrer dans ce qui suit.

#### *3.3.2.1. Notations et terminologie*

Soit $G = (\boldsymbol{V}, \boldsymbol{E}_{>\theta})$ le graphe seuil supérieur, avec $V = \{v_1, v_2, \dots, v_n\}$ est l'ensemble de ses sommets et $\boldsymbol{E}_{>\theta}$ l'ensemble de ses arêtes. Dans ce qui suit, en supposant que les sommets de $G$ sont colorés, les notations utilisées sont les suivantes :

- $\Delta$ : le degré maximum de $G$.
- $c(v_i)$ : la couleur (nombre entier) du sommet $v_i$ dans $G$.
- $N(v_i)$ : le voisinage du sommet $v_i$ dans $G$.
- $N_c(v_i)$ : le voisinage en couleurs du sommet $v_i$ défini par l'ensemble des couleurs ayant au moins un sommet adjacent à $v_i$.
- $L$ : l'ensemble des couleurs utilisées pour colorer $G$.
- $D_m$ : l'ensemble des couleurs qui ont au moins un *sommet dominant*.
- $ND_m$ : l'ensemble des couleurs n'ayant aucun *sommet dominant* ($i.e.$ $ND_m = L \setminus D_m$).
- $dist(v_i, c)$ : la distance entre un sommet $v_i$et une couleur $c$. Elle est définie en se basant sur la notion de *lien minimum* comme la distance entre $v_i$ et le sommet le plus proche coloré avec $c$.

$$dist(v_i, c) = min\{d(X_i, X_j) \mid 1 \leq i \neq j \leq n \ et \ c(v_j) = c\} \tag{3.3}$$

#### *3.3.2.2. Les différentes étapes de l'algorithme*

Les données à grouper sont maintenant résumées dans un graphe non-complet et pondéré $G = (\boldsymbol{V}, \boldsymbol{E}_{>\theta})$. L'objectif est de diviser l'ensemble des sommets $V$ dans une partition $P$ en $k$ classes $\{C_1, C_2, \dots, C_k\}$ où $\forall\, C_i, C_j \in P, C_i \cap C_j = \emptyset \ pour \ tout \ i \neq j$ (le nombre de classes $k$ n'étant pas fixé à l'avance). La notation $P$ est employée pour représenter aussi bien l'ensemble de clusters que l'ensemble de couleurs dans $G$, étant donné que la notion de couleur est équivalente à la notion de classe avec notre approche de classification automatique par *b-coloration* de graphes. La construction d'une *b-coloration* des sommets du graphe $G$ s'effectue en deux étapes : 1) générer une *coloration propre* des sommets de $G$ avec un nombre maximum de couleurs, et 2) supprimer, par une procédure gloutonne, chacune des couleurs n'ayant pas de sommet dominant jusqu'à stabilité de la coloration (*i.e.* où toutes les couleurs du graphe $G$ sont dominantes).

### *a. Première étape, une coloration propre de G avec un maximum de couleurs*

Afin de faciliter la compréhension de l'algorithme, les routines suivantes sont proposées :

- $Mise\_à\_jour(N_c(v_i))$ est une méthode qui effectue une mise à jour de l'ensemble des couleurs voisines à un sommet $v_i$ $(i.e. N_c(v_i))$ en examinant les couleurs de ses sommets adjacents quand celles-ci ont été changées.
- $Ajout\_trié(x, S)$ est une méthode qui ajoute le sommet $x$ à l'ensemble de sommets $S$ en respectant l'ordre décroissant des degrés de ses sommets.
- $Ajout(x, S)$ est une méthode qui ajoute le sommet (*resp.* couleur) $x$ à l'ensemble de sommets (*resp.* couleurs) $S$.
- $Suppression(x, S)$ est une méthode qui supprime un élément $x$ d'un ensemble $S$.

Considérons une autre notation $T$ pour représenter l'ensemble des sommets colorés. Cet ensemble est trié par ordre décroissant en fonction des degrés de ses sommets. Initialement, étant donné que les sommets de $G$ ne sont pas encore colorés, $T$ est vide ($\emptyset$). Il sera mis à jour durant la première phase de l'algorithme de *b-coloration*.

La première procédure *Initier_b-coloration* fournit une première configuration en utilisant le nombre maximum de couleurs disponibles pour la *b-coloration* (équation 3.1) (*i.e.* $\Delta + 1$). Quant les sommets ne sont pas encore colorés, elle commence par le premier sommet de $V$ avec le degré maximum $\Delta$ (notons par v soit un tel sommet). L'algorithme fixe alors $c(v) = 1$ et ajoute $v$ à l'ensemble $T$. La procédure essaye ensuite de colorer les autres sommets selon le principe suivant : pour chaque sommet $v_i$ appartenant à $T$ (initialement $T = \{v\}$), une nouvelle couleur est assignée à chacun de ses voisins $v_j$ si il n'est pas déjà coloré et qui sera à son tour rajouté à $T$. Afin de générer une coloration propre de $G$ avec un nombre maximum de couleurs, la couleur de $v_j$ devrait être différente de celles de ses voisins et aussi de la couleur des voisins de $v_i$. Si toutes les couleurs disponibles pour la *b-coloration* de $G$ ne peuvent pas vérifier pas ces deux contraintes, c'est au moins la première contrainte qui doit être satisfaite (*i.e.* la couleur de $v_j$ doit être différente de ceux de ses voisins). Dans ce cas, comme notre objectif est de trouver une partition où la somme de dissimilarités entre les individus d'une même classe est réduite au minimum, la couleur dont la distance avec $v_j$ est minimale sera sélectionnée, si nous devons choisir entre plusieurs couleurs pour le sommet $v_j$.

Après la coloration de chaque sommet $v_j$ voisin de $v_i$, la procédure vérifie si la couleur du sommet $v_i$ choisi initialement $(c(v_i))$ est devenue dominante. Le sommet $v_i$ est par la suite retiré de $T$.

L'algorithme proposé est donc le suivant :

**Entrée**: *Un graphe seuil supérieur G*
**Sortie**: *Une coloration initiale de G avec un maximum de couleurs*

1: *sélectionner le sommet v avec le degré maximum* Δ;
2: $c(v) \leftarrow 1$;
3: **Ajout**$(v, T)$;
4: $L \leftarrow \{1,2, \dots, \Delta + 1\}$;
5: **répéter**
6: *sélectionner le sommet* $v_i$ *de* $T$;
7: $M \leftarrow N_c(v_i) \cup c(v_i)$;
8: $q \leftarrow 0$;
9: **pour tout** *sommet* $v_j \in N(v_i)$ *tel que* $c(v_j) = \emptyset$ **faire**
10: $q \leftarrow min\{h | h > q, h \notin M \text{ et } h \notin N_c(v_j)\}$;
11: **si** $q \leq \Delta + 1$ **alors**
12: $c(v_j) \leftarrow q$;
13: **sinon**
14: $H \leftarrow \{h | h \in L \text{ et } h \notin N_c(v_j)\}$;
15: $c(v_j) \leftarrow argmin_{h \in H}\left(dist(v_j, h)\right)$;
16: **fin si**
17: **Ajout_trié**$(v_j, T)$;
18: **pour tout** *sommet* $v_h \in N(v_j)$**faire**
19: **Ajout** $\left(c(v_j), N_c(v_h)\right)$;
20: **fin pour**
21: **fin pour**
22: **si** $N_c(v_i) = L \setminus c(v_i)$ **alors**
23: **Ajout**$(c(v_i), D_m)$;
24: **fin si**
25: **Suppression**$(v_i, T)$;
26: **jusqu'a** $(T = \emptyset)$

**Algorithme 1 :** La procédure *Initier_b-coloration*

### *Application de la procédure sur un exemple simple*

Soit l'ensemble d'individus {**A**, **B**, **C**, **D**, **E**, **F**, **G**, **H**, **I**} relatif à la matrice des dissimilarités $D$ de du tableau 3.1.

La procédure *Initier_b-coloration*, appliquée au graphe seuil supérieur $G_{>0.15}$ de la figure 3.1, commence par le sommet **B** ($i.e.$ T = {**B**}) avec un degré égal à 6 (c'est le degré maximum du graphe $G_{>0.15}$). Elle fixe la couleur de **B** à 1 ($i.e.\ c(\mathbf{B}) = 1$) =1 et attribue une nouvelle couleur à chacun de ses voisins. En conséquence, elle génère le graphe coloré de la figure 3.2.

L'ensemble $T$ contient maintenant les sommets {**F**, **A**, **D**, **E**, **C**, **H**} triés par ordre décroissant de degré. La procédure essaye de colorer les sommets restants (**G** et **I**). Soit le sommet **F** de $T$, une nouvelle couleur (la couleur **7**) est attribuée à son voisin **G** qui est différente de celles de son voisinage et du voisinage de **F**. La figure 3.3 montre la configuration retournée.

En conséquence, T comporte maintenant les sommets {**A**, **D**, **E**, **C**, **H**, **G**}. Considérons le sommet **A** de $T$, une nouvelle couleur (la couleur **3**) est attribuée à son sommet voisin **I** qui

est différente de celles de son voisinage et du voisinage de **A**. Ainsi, la configuration initiale utilisant le nombre maximum de couleurs disponibles pour la *b-coloration* de $G_{>0.15}$ (c.-à-d. Δ+1=7) est illustrée sur la figure 3.4. Parmi les sept couleurs de $G_{>0.15}$, il y a seulement deux couleurs qui sont dominantes (la couleur **1** et la couleur **6**) qui possèdent chacune un sommet dominant (**B** pour la couleur **1** et **F** pour la couleur **6**). En conséquence, $D_m = \{1,6\}$.

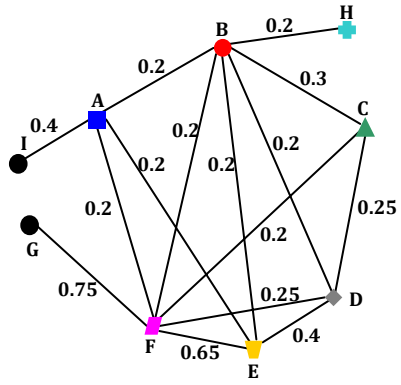



**Figure 3.2 - Coloration du graphe $G_{>0.15}$ (Utilisation du sommet B)**

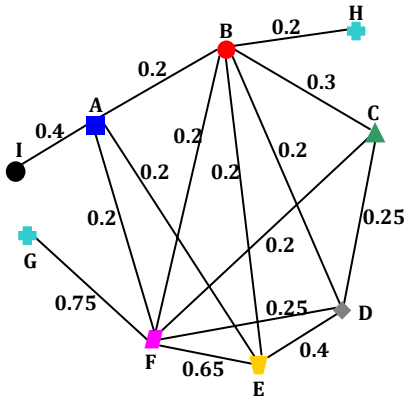



**Figure 3.3 - Une nouvelle Coloration du graphe $G_{>0.15}$ (Utilisation du sommet F)**

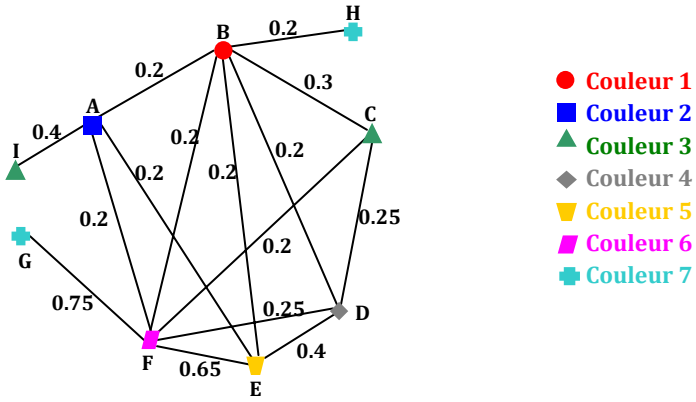



**Figure 3.4 - Une nouvelle Coloration du graphe $G_{>0.15}$ (Utilisation du sommet A)**

***Proposition 3.2*** *La complexité de la procédure Initier_b-coloration est de l'ordre de O($n^2\Delta$).*

***Preuve*** *La procédure Initier_b-coloration est appliquée une fois à chaque sommet coloré de T (au maximum n). Elle commence par attribuer une couleur à chacun des voisins non colorés de ce sommet (au maximum Δ). Chaque voisin est ensuite inséré dans T en respectant l'ordre décroissant des degrés de ses sommets (au maximum en O(n)) et le changement de sa couleur est propagé à ses propres voisins (au maximum Δ). Ainsi, la procédure Initier_b-coloration utilise au maximum (n*Δ*(n+Δ)) instructions : la complexité est de* $O(n^2\Delta)$.

### *b. Deuxième étape, vers une b-coloration de G*

Après exécution de la première procédure *Initier_b-coloration*, certaines couleurs demeurent sans aucun sommet dominant. La procédure *Recherche_b-coloration* suivante cherche une *b-coloration* du graphe $G$ où toutes les couleurs appartenant de $L$ sont dominantes (*i.e.* $D_m = L$). Son principe est le suivant : chaque couleur $q$ de $ND_m$ peut être changée (*c.f. Proposition* 3.3). En effet, après avoir retiré $q$ du graphe $G$, chaque sommet $v_i$ coloré avec $q$ (*i.e.* $c(v_i) = q$), une nouvelle couleur différente de celles de ses voisins (*i.e.* n'appartenant pas à $N_c(v_i)$) lui est attribuée. Etant donné que notre objectif est de trouver une partition où la somme de dissimilarités entre les individus d'une même classe est réduite au minimum, la couleur dont la distance avec $v_i$ est minimale sera sélectionnée, si nous devons choisir entre plusieurs couleurs pour le sommet $v_i$.

Une fois tous les sommets $v_i$ coloré initialement avec $q$ étant recolorés et avant de recommencer avec une autre couleur *non dominante* $q' \in ND_m$, la procédure *Recherche_b-coloration* vérifie si certaines couleurs *non dominantes* de $ND_m$ sont devenues *dominantes* (dans ce cas, ces couleurs sont ajoutées à l'ensemble $D_m$).

L'algorithme correspondant est donc le suivant :

***Entrée***: *Le graphe G coloré généré par la procédure* $Initier\_b-coloration$
***Sortie***: *Une* $b-coloration$ *de G*

1: ***répéter***
2: $l \leftarrow max\{h | h \in ND_m\}$;
3: $L \leftarrow L \setminus \{l\}$;
4: $ND_m \leftarrow L \setminus D_m$;
5: ***pour tout*** *sommet* $v_i$ *tel que* $c(v_i) = l$ ***faire***
6: $H \leftarrow \{h | h \in L \text{ et } h \notin N_c(v_i)\}$;
7: $c(v_i) \leftarrow argmin_{h \in H}(dist(v_i, h))$;
8: ***fin pour***
9: ***pour tout*** *sommet* $v_j$ *tel que* $c(v_j) \in ND_m$ ***faire***
10: ***Mise_à_jour*** $(N_c(v_j))$;
11: ***si*** $N_c(v_j) = L \setminus c(v_j)$ ***alors***
12: ***Ajout***$(c(v_j), D_m)$;
13: ***fin si***
14: ***fin pour***
15: ***jusqu'a*** $(ND_m = \emptyset)$

**Algorithme 2 :** La procédure *Recherche_b-coloration*

Il est clair que la coloration produite par la procédure *Recherche_b-coloration* est une coloration propre. Ceci suit du fait que quand la couleur $c(v_i)$ est changée, sa nouvelle couleur est choisie de telle sorte qu'elle soit différente des couleurs de ses voisins.

La procédure *Recherche_b-coloration* s'achève quand l'ensemble des couleurs *non dominantes* devient vide. En conséquence, il existe au moins un sommet dominant pour chacune des couleurs restantes dans le graphe.

***Proposition 3.3*** *Chaque couleur non dominante q peut être facilement changée*

***Preuve*** *Une couleur non dominante q est une couleur sans sommets dominants. Ainsi, chaque sommet v de couleur q n'est pas adjacent à toutes les couleurs de G. La couleur de v peut être alors changée. En conséquence, la couleur q peut être facilement retirée de G.*

***Proposition 3.4*** *La procédure Recherche_b-coloration génère une b-coloration de G en* $O(n\Delta^2)$.

***Preuve*** *La procédure Recherche_b-coloration est appliquée pour chaque couleur q sans sommets dominants (au maximum Δ). La couleur q est retirée du graphe et chaque sommet* $v_i$ *coloré avec q (au maximum n) est recoloré. Suite à ces transformations, chaque sommet* $v_j$ *tel que* $c(v_j) \in ND_m$ *(au maximum n), examine les couleurs de ses voisins (au maximum Δ) pour mettre à jour son voisinage en couleurs* $N_c(v_j)$. *La dominance de sa couleur* $c(v_j)$ *est ensuite vérifiée. Ainsi, la procédure Recherche_b-coloration utilise au maximum (Δ*(n+n*Δ)) instructions : la complexité est de l'ordre de* $O(n\Delta^2)$.

***Application de la procédure sur l'exemple***

Continuons avec l'exemple précédent où $L$ est l'ensemble des sept couleurs employées pour la coloration de $G_{>0.15}$ $(i.e. L = \{1,2,3,4,5,6,7\})$ $D_m = \{1,6\}$ et $ND_m = \{2,3,4,5,7\}$, la procédure *Recherche_b-coloration* essaye de changer chaque couleur *non dominante*.

Initialement, la dernière couleur affectée 7 est retirée du graphe $G_{>0.15}$. Pour chaque sommet coloré avec 7 (*i.e.* **H** puis **G**), la procédure *Recherche_b-coloration* attribue une nouvelle couleur pour **H** et puis pour **G**. Ces nouvelles couleurs doivent être différentes de celles de leurs voisinages. Il y a cinq couleurs possibles {2,3,4,5,6} pour **H** et également pour **G** {1,2,3,4,5}. Selon la formule 3.3, la couleur choisie pour **H** est celle ayant la distance minimale avec **H**, à savoir ici la couleur 5 ($d(\mathbf{H},\mathbf{E}) = 0.05$ est le minimum et $c(\mathbf{E}) = 5$). Pour le sommet **G**, la couleur choisie est également la couleur 5 ($d(\mathbf{G},\mathbf{H}) = 0.05$ est le minimum et $c(\mathbf{H}) = 5$). La figure 3.5 fournit une illustration du nouveau graphe coloré.

Avant de repartir sur une autre couleur *non dominante* $q' \in ND_m$, nous vérifions si certaines couleurs *non dominantes* de $ND_m$ sont devenues *dominantes* : il résulte que $D_m = \{1,6\}$ et $ND_m = \{2,3,4,5,\}$.

Ensuite une autre couleur de $ND_m$ est prise (ici la couleur 5). Le principe précédent est appliqué aux sommets concernés : **E**, **H** et **G**. La nouvelle couleur pour **E** est 3 (c'est la seule couleur différente des couleurs de son voisinage). La nouvelle couleur pour **H** et **G** est également la couleur 3 (voir figure 3.6). $D_m$ devient l'ensemble {1,6,3} et $ND_m = \{2,4\}$.

De nouveau, une autre couleur de $ND_m$ est choisie (c'est la couleur 4 cette fois), qui concerne seulement le sommet **D**. La couleur de **D** est changée par la couleur 2 vu que c'est la seule couleur différente des couleurs de son voisinage. Ainsi, $D_m$ devient l'ensemble {1,6,3,2} et $ND_m$ l'ensemble vide. En conséquence, la *b-coloration* du graphe $G_{>0.15}$, retournée par la procédure *Recherche_b-coloration,* est donnée dans la figure 3.7.

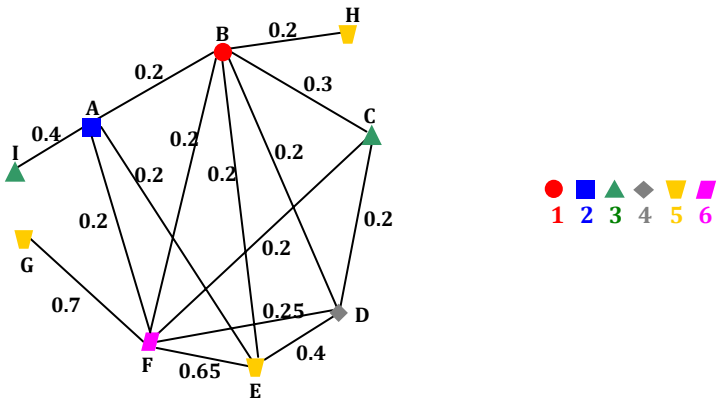



**Figure 3.5 - Une nouvelle coloration du graphe G>0.15 (la couleur 7 est retirée)**

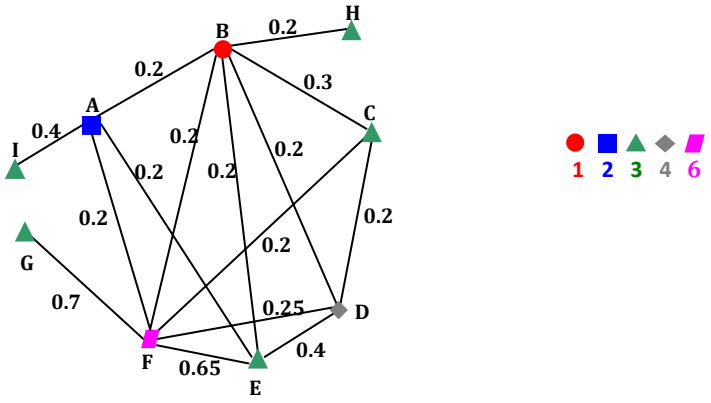



**Figure 3.6 - Une nouvelle coloration du graphe $G_{>0.15}$ (la couleur 5 est retirée)**

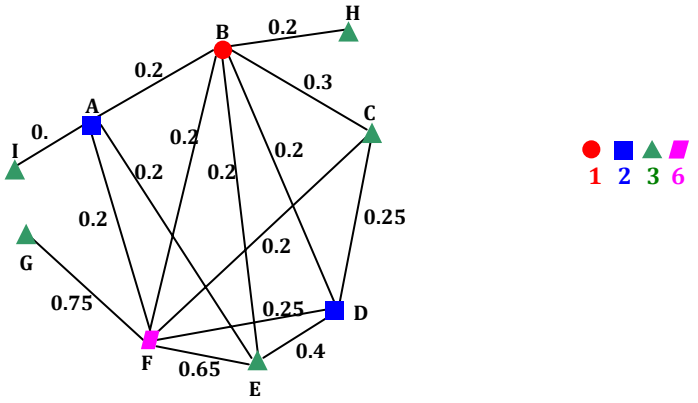



**Figure 3.7 - La b-coloration du graphe $G_{>0.15}$ (quatre couleurs sont utilisées)**

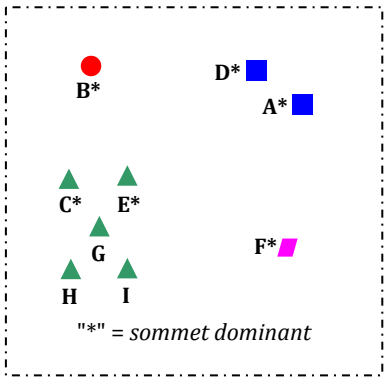



**Figure 3.8 - La partition associée au seuil θ =0.15**

Ainsi, la *b-coloration* du graphe $G_{>0.15}$ (voir figure 3.8) induit la partition en quatre classes (chaque couleur correspond à une classe) : $C_1 = \{\mathbf{B}\}$, $C_2 = \{\mathbf{A}, \mathbf{D}\}$, $C_3 = \{\mathbf{C}, \mathbf{E}, \mathrm{G}, \mathrm{H}, \mathrm{I}\}$ et $C_4 = \{\mathbf{F}\}$. Les sommets avec la même couleur (forme géométrique) sont groupés dans la même classe et les lettres en gras représentent les sommets dominants. Ceci signifie que ces sommets sont adjacents à au moins un sommet dans chacune des autres classes de couleurs.

***Proposition 3.5*** *L'algorithme de b-coloration avec les deux procédures Initier_b-coloration et Recherche_b-coloration génère une b-coloration de tout graphe G.*

***Preuve*** *Supposons que la coloration de G n'est pas une b-coloration. Par conséquent, la coloration de G n'est pas une coloration propre ou bien il existe au moins une couleur c sans aucun sommet dominant. Supposons que la coloration de G n'est pas une coloration propre. Ceci implique qu'il existe au moins deux sommets adjacents ayant la même couleur. Cependant, tant que les procédures Initier_b-coloration et Recherche_b-coloration fonctionnent, la couleur de chaque sommet est choisie de telle sorte qu'elle soit différente de celles de ses voisins. Cela nous prouve bien que la coloration de G est propre. D'où la contradiction. Comme la proposition 3.3 prouve que chaque couleur non dominante peut être facilement changée, la procédure Recherche_b-coloration s'achève quand l'ensemble de*

*couleurs sans sommets dominants devient vide et induit ainsi une coloration dominante de G. Ceci contredit la prétention « il existe au moins une couleur c sans aucun sommet dominant ». En conséquence, il est évident qu'en utilisant notre algorithme, un graphe quelconque devrait avoir une b-coloration,*

***Proposition 3.6*** *Chaque classe produite par l'algorithme de classification automatique par b-coloration a un diamètre égal à 1 dans le complément du graphe* $G = (\boldsymbol{V}, \boldsymbol{E}_{>\theta})$.

***Preuve*** *Par la propriété de coloration propre, nous dégageons facilement que pour chaque paire de sommets adjacents* $(v_i, v_j) \in E_{>\theta}, c(v_i) \neq c(v_j)$. *Par conséquent, il n'y a aucune arrête entre les sommets d'une même couleur. Chaque classe est vue donc comme ensemble indépendant (Independent set en anglais). En prenant le complément du graph G (c.-à-d. le graphe seuil inférieur dans lequel les sommets correspondent aux individus et les arêtes relient les individus ayant une dissimilarité inférieure au seuil* $\theta$*), chaque classe devient donc un sous-graphe fortement connexe dans laquelle chaque paire de sommets est reliée par une seule arête. En conséquence, le diamètre de chaque classe est égal à 1.*

***Remarque 3.1 :*** Noter que selon la formule 3.2, si le graph seuil supérieur $G$ est un graphe non connexe composé de $p$ composantes connexes $(C_1, C_2, \dots, C_p)$, l'algorithme est appliqué à chacune des composantes séparément afin de trouver la *b-coloration* de *G*.

#### *3.3.2.3. Discussion*

1. A partir de la proposition 3.6, le diamètre de chaque classe produite par l'algorithme de classification par *b-coloration* est égal à 1 dans le graphe de similarité complémentaire de *G*. Ceci est une indication de l'homogénéité des classes obtenues qui est due à leur forte connectivité. En effet, la seule meilleure possibilité en termes de diamètre est que chaque paire de sommets d'une même classe sont reliés par une seule arête.

2. L'idée de base de l'algorithme de classification par *b-coloration* est de partir sur une coloration du graphe avec un nombre maximal de classes (couleurs). Ensuite, chaque couleur non dominante sera modifiée jusqu'à ce que toutes les couleurs de G deviennent dominantes. Avec la procédure *Recherche_b-coloration*, s'il y a plusieurs possibilités pour changer la couleur non dominante d'un sommet, la couleur de la classe la plus proche sera choisie : ceci donne une autre indication sur la forte homogénéité et la bonne séparation des classes obtenues.

3. En outre, il existe au moins un sommet dominant dans chaque classe de couleur. Ce sommet est adjacent à au moins un sommet dans chacune des autres classes : il reflète les propriétés de sa classe mais garantie également une séparation nette de la classe vis-à-vis des autres classes de la partition obtenue.

4. Et pour conclure, nous rappelons que notre objectif n'est pas d'obtenir une *b-coloration* optimale (avec le nombre optimal de couleurs), mais plutôt un meilleur partitionnement des données se basant sur les bonnes propriétés de *la b-coloration de graphes* et en temps d'exécution raisonnable.

### 3.3.3. Choix du seuil de dissimilarité

La méthodologie de classification non supervisée par *b-coloration* de graphes que nous avons proposée est itérative. Elle consiste, à chaque itération correspondante à un seuil de dissimilarité extrait de manière croissante de la matrice de dissimilarités *D*, à :

- *appliquer l'algorithme de b-coloration décrit ci-dessus sur le graphe seuil supérieur en question.*
- *évaluer la qualité des partitions obtenues (i.e. une pour chaque seuil) en utilisant un critère spécifique d'optimalité (section 2.5) afin d'identifier la meilleure partition qui sera renvoyée à l'utilisateur.*

A titre d'illustration, pour chaque seuil de dissimilarité du tableau 3.1, le graphe seuil supérieur a été construit, l'algorithme de classification par *b-coloration* est ensuite appliqué pour produire les partitions correspondantes avec des valeurs de l'indice de *Dunn généralisé* ($Dunn\_gen$) différentes (voir tableau 3.2). La partition $\theta = 0.15$ produit ainsi la valeur maximale de $Dunn\_gen$ (1.52) par rapport aux autres partitions avec des seuils θ différents.

| $\boldsymbol{Seuil\ \theta}$ | $\boldsymbol{Partition\ P}$ | $\boldsymbol{\#classes\ k}$ | $\boldsymbol{Dunn\_gen(P)}$ |
|---|---|---|---|
| 0.05 | {**A**}{**B**}{**C**}{**D**}{**E**,H}{**F**}{**G**}{**I**} | 8 | 1.50 |
| 0.075 | {**A**}{**B**,**I**}{**C**}{**D**}{**E**,H}{**F**}{**G**} | 7 | 1.00 |
| 0.10 | {**A**}{**B**,**I**}{**E**,C}{**G**,D}{**F**,H} | 5 | 1.25 |
| 0.15 | {**A**,**D**}{**B**}{**C**,**E**,G,H,I}{**F**} | 4 | 1.52 |
| 0.20 | {**D**,A,B,H}{**E**,C,G,I}{**F**} | 3 | 1.16 |
| 0.25 | {**A**,**B**,**E**,**G**,H}{**C**,**D**,**F**,**I**} | 2 | 1.15 |
| 0.30 | {**A**,**E**,**G**,B,C,H}{**D**,**F**,**I**} | 2 | 1.30 |
| 0.40 | {**F**,A,B,C,D,H,I}{**E**,**G**} | 2 | 1.32 |
| 0.65 | {**F**,A,B,C,D,E,H,I}{**G**} | 2 | 1.01 |

**Tableau 3.2 - Evaluation des partitions obtenues**

## 3.4. Expérimentations et performances

### 3.4.1. Introduction

Notre méthode de classification par *b-coloration* de graphes permet bien de traiter aussi bien les données de *descriptions classiques* que les données de *descriptions symboliques*, dès lors qu'un tableau de dissimilarités peut être construit sur les données. Dans ce chapitre, nous allons présenter quelques applications de la méthode et évaluer ses

performances. Nous avons d'abord appliqué la méthode sur des jeux de données classiques (*benchmarks*) issues de la base UCI, à savoir *"Zoo", "Auto Import", "Mushroom», "Wine"* et *"Heart Disease Databases"* [Blake and Merz, 1998]. Nous avons ensuite évalué l'approche sur des échantillons de données de natures complexes (décrits à la fois par des variables *univaluées* et *multivaluées*) extraits d'une base de données PMSI-2003 fournie par l'Agence Régionale d'Hospitalisation Rhône-Alpes (ARH-RA). Celle-ci recense les données PMSI de tous les établissements de santé de la Région (publics et privés). L'objectif principal étant de fournir une partition fine d'un ensemble de séjours en groupes homogènes et bien séparés et d'identifier les problèmes de la classification actuelle en *Groupes Homogènes de Malades*. Nous avons finalement appliqué notre méthode sur un jeu *d'images archéologiques* annotées, illustrant ainsi les possibilités de la méthode.

Sur ces données, la méthode est comparée à une variété d'approches de classification automatique, à savoir l'approche symbolique de classification ascendante hiérarchique de Mali et Mitra et la *classification conceptuelle*, toutes les deux présentés dans [Mali and Mitra, 2003], l'approche de classification ascendante hiérarchique avec ses différentes variantes (*saut minimum*, *saut maximum, saut moyen* et *saut barycentrique*), l'approche de classification de Hansen basée également sur une technique de *coloration minimale de graphes* [Hansen and Delattre, 1978], l'algorithme ROCK [Guha *et al.,* 2000] et l'algorithme populaire de *k-moyennes* [Hartigan and Wong, 1979; Mac-Queen, 1967].

Malgré leurs différences algorithmiques significatives, les algorithmes de classification avec lesquels nous nous sommes comparés ont des propriétés communes avec l'approche par *b-coloration* de graphes. Même si cette dernière n'est pas une méthode hiérarchique, elle nécessite pour sa mise en œuvre la définition d'une matrice de dissimilarités entre les individus et peut être ainsi applicable à tout type de données. Ceci étant le cas pour les approches de classification ascendante hiérarchique et l'approche de Hansen. Cette dernière approche partage également une autre propriété avec l'approche proposée dans le fait qu'elle est fondée sur une technique de coloration de graphes baptisée la *coloration minimale*. Enfin, la nouvelle approche de classification par *b-coloration* est qualifiée comme une méthode par partitionnement. C'est pour cette raison qu'elle sera comparée à l'approche *k-moyennes* quat nous sommes en présence de données numériques (pour la base *"Wine"* par exemple).

### 3.4.2. Critères de comparaison de deux partitions

Pour une meilleure évaluation des résultats obtenus par les différentes approches de classification automatique, nous avons utilisés les deux critères suivants pour comparer deux partitions *P* et *P'* d'un même ensemble d'individus.

- Deux mesures d'appariement probabilistes connues sous le nom *cohésion* et *distinction* [Biswas *et al.*, 1998] Elles sont liées respectivement à la *similarité intraclasse* et la *dissimilarité intraclasse*.

- Une mesure statistique d'appariement de labels appelé *l'indice de Rand* [Rand, 1971]. Il calcule la similarité entre deux partitions afin d'évaluer *l'exactitude, la pureté* ou *l'accord* de la classification obtenue par rapport à un partitionnement espéré. Dans la suite, ce critère est utilisé uniquement pour les jeux de données qui sont déjà étiquetés (à savoir *"Zoo"*, *"Wine"* et *"Heart Disease Databases"*).

#### *3.4.2.1. Les fonctions de cohésion et de distinction*

Comme dans [Mali and Mitra, 2003], nous avons adoptés les deux fonctions de *cohésion* et de *distinction* dans notre problème de comparaison de partitions. Elles mesurent respectivement le degré de *similarité intraclasse* et de *dissimilarité intraclasse* en se basant uniquement sur les valeurs prises par les attributs décrivant les individus dans les classes et non pas sur les ressemblances entre les individus. De telles fonctions fonctionnent indépendamment du nombre de classes et des dissimilarités entre les individus : elles c'est sont ainsi très appropriées pour le problème d'évaluation de clustering. D'autre part, elles peuvent être utilisées avec des données hétérogènes.

Pour une partition $P$ en $k$ classes $\{C_1, C_2, \dots, C_k\}$ de l'ensemble d'individus $X = \{X_1, X_2, \dots, X_n\}$, la *cohésion* est employée pour refléter la compacité des classes découvertes. Elle est mesurée comme l'accroissement dans le nombre attendu de modalités des variables qui prédisent correctement la classe par rapport au nombre de modalités correctes sans connaissance d'une telle classe.

L'accroissement normalisé dans cette prédictibilité pour un individu $v$ appartenant à une classe $C_h$ noté $M_{vh}$, mesure la prévisibilité d'appartenance de $v$ à la classe $C_h$. Il est défini comme suit :

$$M_{vh} = \frac{1}{\sum_{i=1}^{m} |Y_i(v)|} \sum_{i=1\dots m, j \in Y_i(v)} \left( \left( Prob\left(Y_i = V_{ij} \middle| C_h\right) \right)^2 - \left( Prob\left(Y_i = V_{ij}\right) \right)^2 \right) \qquad (3.4)$$

où $m$ est le nombre de variables caractérisant les individus. $|Y_i(v)|$ est le nombre de valeurs prises par la variable $Y_i$ pour décrire l'individu $v$. $Prob\left(Y_i = V_{ij}\right)$ est la probabilité que la variable $Y_i$ prend la valeur $V_{ij}$. $Prob\left(Y_i = V_{ij} \middle| C_h\right)$ est la probabilité conditionnelle que la variable $Y_i$ prend la valeur $V_{ij}$ dans la classe $C_h$.

Cette équation suppose que l'individu $v$ ne prend qu'une seule valeur par variable (représentée par $j \in Y_i(v)$).

La valeur normalisée de la cohésion est donnée par la moyenne des valeurs $M_{vh}$ calculées pour tous les individus de la population. Ceci peut être interprété comme l'accroissement dans l'appariement entre un individu et sa classe dans la partition par rapport à l'appariement entre l'individu et l'ensemble total de données [Biswas *et al.*, 1998].

$$cohésion(P) = \frac{\sum_{h=1}^{k} \sum_{v \in C_h} M_{vh}}{n} \quad (3.5)$$

où $n$ est le nombre d'individus de la population. Plus grande étant cette valeur de *cohésion*, plus sont compactes les classes de la partition $P$.

La *distinction* est définie comme la *dissimilarité intraclasse* utilisant une mesure d'appariement probabiliste entre les classes, appelée la *variance d'appariement de distribution*. Ce dernier, calculé entre deux classes $C_h$ et $C_l$ de$P$, est donné par:

$$Var(C_h, C_l) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j \in Y_i} \left( Prob\left(Y_i = V_{ij} | C_h\right) - Prob\left(Y_i = V_{ij} | C_l\right) \right)^2 \quad (3.6)$$

Plus grande étant cette valeur de *distinction*, plus sont dissimilaires les deux classes comparées et ainsi les concepts qu'elles représentent.

La *distinction* de la partition P est donnée par la variance moyenne entre les classes de la partition.

$$distinction(P) = \frac{\sum_{h=1}^{k} \sum_{l=1}^{k} Var(C_h, C_l)}{k(k-1)} \quad (3.7)$$

Quant nous comparons deux partitions, celle qui produit la plus grande distinction est la préférée puisque les clusters de cette partition représentent des concepts plus distincts [Biswas *et al.*, 1998].

#### *3.4.2.2. L'indice de Rand*

Dans notre cas, certaines bases utilisées de l'UCI (à savoir *"Zoo"*, *"Wine"* et *"Heart Disease Databases"*) incluent des informations sur les classes recherchées (étiquettes). Ces étiquettes sont disponibles pour l'évaluation mais non visibles à l'algorithme de classification. L'objectif étant d'effectuer une classification non supervisée qui identifie correctement les structures cachées dans les données, nous utilisons ainsi les étiquettes de classe uniquement dans l'étape d'évaluation. En conséquence, notre évaluation sera basée sur un indice d'accord appelée *l'indice de Rand*. Il permet d'estimer l'exactitude de classification en calculant le pourcentage global de paires en accord.

La partition retournée par une approche de classification automatique est considérée comme une relation entre les individus de la population $X$ : pour chaque paire d'individus, ils ont soit la même étiquette (classe), soit des étiquettes différentes. Pour une population de $n$ individus, il y'a $n(n-1)/2$ paries d'individus $(X_i, X_j)$, et ainsi $n(n-1)/2$ couples de décisions différentes. L'indice de Rand est donné par :

$$Rand(P, P') = exactitude = \frac{n_1 + n_2}{n(n-1)/2} \quad (3.8)$$

où :

- $P$ est la partition produite par l'algorithme de classification automatique utilisé.

- $P'$ est la *vraie* partition (ou aussi *estimée* ou *recherchée*). Elle est donnée par les étiquettes prédéfinies des classes.
- $n_1$ et $n_2$ donnent le nombre total *d'accords* ou de *décisions correctes* entre les deux partitions $P$ et $P'$. $n_1$ fournit le nombre d'accords positifs où $X_i$ et $X_j$ appartiennent à la même classe dans $P$ et $P'$. $n_2$ fournit le nombre d'accords négatifs où $X_i$ et $X_j$ sont dans des classes différentes dans $P$ et $P'$.

Quand nous comparons deux approches de classification, celle qui produit la plus grande exactitude est la préférée puisque les clusters de cette partition identifient correctement les informations cachées dans les données.

### 3.4.3. Jeux de données classiques de l'UCI

Dans les applications présentées ici, la distance euclidienne a été appliquée pour définir le degré de dissimilarité entre les individus (équation 2.27).

Les résultats des différentes approches de classification automatique sur les jeux de données classiques (*benchmarks*) de l'UCI, présentées dans cette section, ont été obtenus comme suit :

- Concernant l'approche symbolique de classification ascendante hiérarchique et la *classification conceptuelle*, les résultats ont été retirés de [Mali and Mitra, 2003] et ne sont pas reproduis ici. C'est pour cette raison que les comparaisons avec ces deux approches sont uniquement basées sur les critères de *cohésion* et la *distinction*.
- Pour l'algorithme *ROCK* et l'approche de *classification ascendante hiérarchique* utilisant un *saut barycentrique*, les résultats sont obtenues de [Guha *et al.,* 2000].
- Par ailleurs, les autres approches (à savoir, la *Classification Ascendante Hiérarchique CAH*, *l'approche Hansen*, les *k-moyennes*) ont été implémentées et la partition optimale est identifiée pour chacune d'elles en utilisant les deux indices de validité de *Davies-Bouldin* et de *Dunn généralisé* selon le principe suivant : comme mentionné dans la section 2.5, une partition de bonne qualité correspond à une petite valeur de l'indice de *Davies-Bouldin* et à une grande valeur de l'indice de *Dunn généralisé*. En pratique, plusieurs valeurs du nombre de classes (ou du seuil de dissimilarité pour l'approche de Hansen) sont fixées et les partitions correspondantes sont comparées en utilisant d'une part l'indice de *Davies-Bouldin* et d'autre part l'indice de *Dunn généralisé*. Nous calculons par la suite, pour les deux partitions optimales résultantes, les valeurs de *cohésion* et de *distinction*. La partition qui produit les plus grandes valeurs en *distinction* et en *cohésion* sera retenue. Il convient de noter que le logiciel SAS (Statistical Analysis System) a été utilisé pour réaliser l'algorithme de *k-moyennes*.

### *3.4.3.1. La base d'animaux "Zoo"*

Il s'agit d'un jeu de 100 instances d'animaux caractérisées par 17 variables hétérogènes et pré-classés en 7 familles. Le nom d'animal constitue la première variable. Les 15 variables suivantes sont booléennes. Elles sont liées soit à la présence de poils, de plumes, d'œufs, de lait, d'épine dorsale, d'ailerons et de queue chez l'animal, soit à son caractère, si il est aérien, aquatique, prédateur, denté, respire, venimeux, domestique. La dernière variable descriptive est de type quantitatif numérique. Elle précise le nombre de pattes et prend ses valeurs dans l'ensemble {0; 2; 4; 5; 6; 8}.

Le tableau 3.3 suivant fournit les résultats de classification sur la base "Zoo". Nous avons dans un premier temps comparé les méthodes de classification au regard des mesures de distinction et de cohésion. Celles-ci indiquent globalement un meilleur partitionnement pour les sept classes retournées par notre approche de *classification par b-coloration*. Cette dernière réalise également de meilleures performances en termes de critère de Rand (voir tableau 3.4). L'accord de classification par rapport à la partition originale atteint 94.71% pour notre approche alors qu'il vaut 25.35% pour la classification hiérarchique saut minimum, 92.44% pour le saut maximum, 93.21% pour le saut moyen et 84.56% pour l'approche de Hansen. Toutefois, nous pouvons constater que les résultats sont proches de ceux fournis par l'algorithme de classification hiérarchique par *saut moyen* (avec un net avantage pour l'approche par *b-coloration*).

| ***Approche de classification*** | ***#classes k*** | ***Cohésion*** | ***Distinction*** |
|---|---|---|---|
| Approche par b-coloration | 7 | 0.37 | 0.61 |
| Approche symbolique de classification hiérarchique | 4 | 0.27 | 0.58 |
| Classification conceptuelle | 2 | 0.08 | 0.50 |
| CAH (Saut minimum) | 2 | 0.01 | 0.50 |
| CAH (Saut maximum) | 6 | 0.36 | 0.56 |
| CAH (Saut moyen) | 8 | 0.37 | 0.59 |
| Approche Hansen (coloration minimale) | 4 | 0.28 | 0.55 |

**Tableau 3.3 - Performances de la classification sur la base "Zoo"**

| ***Approche de classification*** | ***Exactitude*** |
|---|---|
| Approche par b-coloration | 94.71% |
| CAH (Saut minimum) | 25.35% |
| CAH (Saut maximum) | 92.44% |
| CAH (Saut moyen) | 93.21% |
| Approche Hansen (coloration minimale) | 84.56% |

**Tableau 3.4 - Evaluation de la pureté de classification sur la base "Zoo"**

Dans une perspective de mieux évaluer notre approche, nous avons conduit par la suite des comparaisons supplémentaires avec les trois approches de classification hiérarchique (*saut minimum, saut maximum* et *saut moyen*), quand celles-ci produisent le même nombre de classes que celui de l'approche par *b-coloration* (*i.e.* 7 classes). Les valeurs de cohésion

et de distinction des partitions associées à ces trois approches *saut minimum, saut maximum* et *saut moyen* sont respectivement de (0.26, 0.66), (0.35, 0.55) et (0.34, 0.60), alors que l'accord de classification est estimée respectivement à 78.14%, 92.72% et 89.63%. En conséquence, nous pouvons noter que la partition obtenue par l'approche de *b-coloration* est meilleure que celles produites par les approches de classification hiérarchique, même quand le nombre de classes est identique (saut pour l'approche saut minimum en termes de distinction). A leur tour, les valeurs prises par l'indice de Rand montrent que la partition retournée par l'approche de *b-coloration* identifie correctement la structure en classes cachée dans les données.

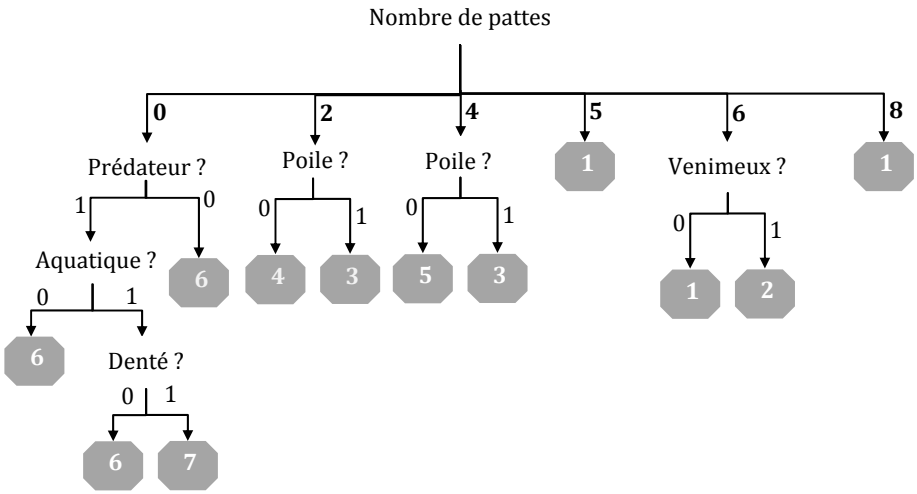



**Figure 3.9 - Représentation des classes d'animaux de la b-coloration par un arbre de décision**

| ***Cluster N°*** | ***Animaux*** |
|---|---|
| 1 | Crayfish; flea; gnat; housefly; ladybird; lobster; moth; octopus; scorpion; starfish; termite |
| 2 | Honeybee; wasp |
| 3 | Aardvark; antelope; bear; boar; buffalo; calf; cavy; cheetah; deer; elephant; fruitbat; giraffe; girl; goat; gorilla; hamster; hare; leopard; lion; lynx; mink; mole; mongoose; opossum; oryx; platypus; polecat; pony; puma; pussycat; raccoon; reindeer; sealion; squirrel; vampire; vole; wallaby; wolf |
| 4 | Chicken; crow; dove; duck; flamingo; gull; hawk; kiwi; lark; ostrich; parakeet; penguin; pheasant; rhea; skimmer; skua; sparrow; swan; vulture; wren |
| 5 | Crab; frog; newt; toad; tortoise; tuatara |
| 6 | Carp; clam; haddock; pitviper; seahorse; seawasp; slowworm; slug; sole; worm |
| 7 | Bass; catfish; chub; dogfish; dolphin; herring; pike; piranha; porpoise; seal; seasnake; stingray; tuna |

**Tableau 3.5 - Effectif par classe pour la base "Zoo"**

La partition optimale à sept classes retournée par l'approche de classification par *b-coloration* est énumérée dans le tableau 3.5 et la figure 3.9. Nous observons bien que les classes résultantes sont sémantiquement significatives sur la base d'un certain nombre de

caractéristiques. La première et la deuxième classe de la partition contiennent des animaux avec plus de cinq pattes. Cependant, la deuxième classe se compose des animaux avec six pattes et peut être distingué de la première sur la base des attributs "venimeux" et la "présence de queue". D'un autre côté, la troisième classe se compose des animaux poilus avec deux ou quatre pattes. Toutefois, la quatrième et la cinquième classe comportent des animaux avec respectivement deux et quatre pattes mais sans présence des poils. Finalement, la sixième et la septième classe se composent des animaux sans pattes bien qu'elles se distinguent sur la base des attributs "prédateur", "aquatique" et "denté". L'analyse de ces propriétés (voir figure 3.9) et de l'effectif par classe (voir tableau 3.5), montrent que l'approche par b-coloration produit bien des classes *pures* et *bien séparées*.

#### *3.4.3.2. La base de maladies du cœur "Heart Disease Databases "*

Il s'agit d'un tableau de données étiquetées (deux classes) et très approprié au problème d'évaluation de clustering en raison de son aspect réel et mixte. Les 303 individus qu'il contient sont 303 maladies du cœur générées par la clinique de Cleveland en 1988. Chaque individu est caractérisé par 13 variables, dont 5 numériques et 8 catégorielles.

Les résultats de classification sont fournis dans le tableau 3.6[3]. Les mesures de cohésion et de distinction montrent que l'approche de *b-coloration* produit le meilleur partitionnement. Néanmoins, l'approche de classification hiérarchique par saut minimum produit des meilleurs résultats que la nôtre selon le critère de distinction, bien qu'elle échoue pour les critères de cohésion et d'exactitude (voir tableau 3.7). Ceci est dû à la mauvaise construction des classes (il convient de noter que cette approche souffre de l'effet de chaîne). En effet, nous avons observé, en analysant les effectifs des classes produites par cette approche, que la taille des classes change considérablement (une classe parmi les trois produites contient 300 maladies).

En outre, les approches hiérarchiques n'atteignent pas les résultats fournis par la b-*coloration* même lorsque nous utilisons le même nombre de classes (*i.e.* 8 classes). La cohésion et la distinction pour les trois algorithmes *saut minimum, saut maximum* et *saut moyen* sont estimés respectivement à (0.02, 0.88), (0.13, 0.41) et (0.1, 0.63).

| ***Approche de classification*** | ***#classes k*** | ***Cohésion*** | ***Distinction*** |
|---|---|---|---|
| Approche par b-coloration | 8 | 0.14 | 0.68 |
| CAH (Saut minimum) | 3 | 0.006 | 0.80 |
| CAH (Saut maximum) | 6 | 0.11 | 0.40 |
| CAH (Saut moyen) | 2 | 0.003 | 0.61 |
| Approche Hansen (coloration minimale) | 3 | 0.05 | 0.45 |

**Tableau 3.6 - Performances de la classification sur la base " Heart Disease Databases"**

[3] Afin de calculer la cohésion et la distinction, l'ensemble des valeurs des attributs numériques sont divisés en cinq parties égales (quintiles).

La pureté de classification s'élève à 58.79% pour l'algorithme de classification par b-coloration de graphes. Autrement, le reste des algorithmes (*i.e. saut minimum, saut maximum*, *saut moyen* et de *Hansen*) réalisent respectivement un accord de classification de 50.35%, de 56.19%, de 50.18% et de 53.36% avec la partition originale prédéfinie en deux classes (voir tableau 3.7).

De plus, l'algorithme de *b-coloration* produit de meilleurs résultats que les approches de *saut minimum, saut maximum* et *saut moyen* quand le nombre de classes est fixé à 8 (*i.e.* le même nombre de classes fournies par la *b-coloration*). Ils réalisent respectivement une pureté de classification de 50.56 %, de 54.86% et de 54.86%.

| ***Approche de classification*** | ***Exactitude*** |
|---|---|
| Approche par b-coloration | 58.79% |
| CAH (Saut minimum) | 50.35% |
| CAH (Saut maximum) | 56.19% |
| CAH (Saut moyen) | 50.18% |
| Approche Hansen (coloration minimale) | 53.36% |

**Tableau 3.7 - Evaluation de la pureté de classification sur la base " Heart Disease Databases "**

### *3.4.3.3. La base de vins "Wine"*

Contrairement aux expériences précédentes, qui se sont servies de jeux de données avec des attributs catégorielles et des approches de classification la plupart hiérarchiques, nous avons également évalué l'approche de classification par *b-coloration* sur un jeu de données de l'UCI avec des attributs purement numériques. Ceci nous a permis de la comparer avec l'algorithme des *k-moyennes* [Hartigan and Wong, 1979; Mac-Queen, 1967].

Il s'agit ici d'un tableau de données numériques, où les 178 instances sont 178 résultats d'une analyse chimique des vins cultivés dans une région spécifique de l'Italie. Trois types de vin sont représentés dans les 178 instances caractérisé chacune par 13 variables chimiques.

Les résultats de classification sont illustrés dans les tableaux 3.8 et 3.9. A partir des critères de cohésion, de distinction et d'exactitude, nous constatons que l'approche de *b-coloration* produit un meilleur partitionnement que les approches de *Hansen* et *k-moyennes*. En outre, en fixant *k*=4 (le même nombre de classes que la *b-coloration*), l'algorithme des *k-moyennes* ne fournit pas de meilleurs résultats, à savoir, 0.09 et 0.37 pour respectivement la cohésion et la distinction et 70.17% pour l'exactitude de classification.

| ***Approche de classification*** | ***#classes k*** | ***Cohésion*** | ***Distinction*** |
|---|---|---|---|
| Approche par b-coloration | 4 | 0.14 | 0.37 |
| Approche Hansen (coloration minimale) | 4 | 0.11 | 0.34 |
| Algorithme k-moyennes | 5 | 0.08 | 0.33 |

**Tableau 3.8 - Performances de la classification sur la base " Heart Disease Databases"**

| *Approche de classification* | *Exactitude* |
|---|---|
| Approche par b-coloration | 83.75% |
| Approche Hansen (coloration minimale) | 74.78% |
| Algorithme k-moyennes | 69.15% |

**Tableau 3.9 - Evaluation de la pureté de classification sur la base " Wine "**

#### *3.4.3.4. La base d'automobiles "Auto"*

Cette base comporte 193 instances d'automobiles décrites par 24 variables, 14 de type quantitatif (à savoir par exemple la longueur de la voiture, sa largeur, sa puissance en chevaux et son prix) et 10 de type qualitatif nominal (à savoir par exemple sa marque, le type de carburant, le nombre de portes, le type de carrosserie et le nombre de cylindres) attributs. Il convient de noter que cette base n'est pas étiquetée (pas de labels de classes).

Le tableau 3.10 illustre les résultats de classification obtenus par les différentes approches de classification sur la base "Auto". D'après les valeurs de cohésion et de distinction, nous pouvons constater que l'approche de classification par *b-coloration* génère des classes plus compactes et bien séparées. Ceci confirme la pertinence de la technique de b-coloration pour offrir un compromis entre la *séparation interclasse* et *l'homogénéité intraclasse*.

Pour une évaluation plus pertinente vis-à-vis les résultats obtenus par les approches de classification hiérarchique, la cohésion et la distinction sont évaluées quand ces approches produisent le même nombre de classes que pour la *b-coloration* (*i.e.* 11 classes). La cohésion et la distinction sont calculées à 0.06, 0.85 (saut minimum) et 0.29, 0.68 (saut maximum). Excepté l'algorithme *saut minimum*, l'approche proposée est meilleure que le reste de méthodes. Néanmoins, l'approche de saut minimum échoue pour la valeur de cohésion et les classes produites par cette approche ne sont pas assez significatives. En effet, 179 des 193 voitures de la base appartiennent à la même classe (problème de l'effet de chaîne).

| *Approche de classification* | *#classes k* | *Cohésion* | *Distinction* |
|---|---|---|---|
| Approche par b-coloration | 11 | 0.33 | 0.75 |
| Approche symbolique de classification hiérarchique | 3 | 0.11 | 0.39 |
| Classification conceptuelle | 2 | 0.07 | 0.33 |
| CAH (Saut minimum) | 3 | 0.01 | 0.64 |
| CAH (Saut maximum) | 12 | 0.26 | 0.70 |
| CAH (Saut moyen) | 11 | 0.27 | 0.72 |
| Approche Hansen (coloration minimale) | 8 | 0.21 | 0.49 |

**Tableau 3.10 - Performances de la classification sur la base " Auto"**

#### *3.4.3.5. La base des champignons "Mushroom"*

Il s'agit d'un autre jeu de données de l'UCI où chaque enregistrement contient l'information qui décrit les 21 propriétés physiques (par exemple, couleur, odeur, taille, forme) d'un champignon. Un enregistrement contient également une étiquette selon que le

champignon est *toxique* ("poisonous") ou *comestible* ("edible"). Toutes ces variables décrivant les champignons sont catégorielles.

La base "*Mushroom*" a le plus grand nombre d'enregistrements (8124) entre les bases de données que nous avons employées dans nos expériences. Le nombre de champignons comestibles et toxiques dans la base est respectivement de 4208 et 3916. Toutefois, Il y a 23 espèces des champignons dans cette base. Chacun de ces espèces est identifié comme comestible, toxique, ou de comestibilité inconnue. Cette dernière classe a été combinée avec la classe toxique.

Le tableau 3.11 ci-dessous fournit les résultats de classification obtenus sur les données "*Mushroom*" par les différentes approches de classification, à savoir la *b-coloration*, *l'approche symbolique de classification hiérarchique*, la *Classification conceptuelle*, *l'approche de Hansen* et les *approches hiérarchiques par saut minimum*, *saut maximum* et *saut moyen*. Nous constatons que la partition optimale renvoyée par notre approche de classification est celle qui apporte le plus en *cohésion* et *distinction*.

| ***Approche de classification*** | ***#classes k*** | ***Cohésion*** | ***Distinction*** |
|---|---|---|---|
| Approche par b-coloration | 17 | 0.41 | 0.71 |
| Approche symbolique de classification hiérarchique | 10 | 0.02 | 0.64 |
| Classification conceptuelle | 4 | 0.01 | 0.57 |
| CAH (Saut minimum) | 20 | 0.21 | 0.62 |
| CAH (Saut maximum) | 21 | 0.32 | 0.65 |
| CAH (Saut moyen) | 22 | 0.34 | 0.67 |
| Approche Hansen (coloration minimale) | 19 | 0.36 | 0.68 |

**Tableau 3.11 - Performances de la classification sur la base " Mushroom"**

| **Approche de classification hiérarchique saut barycentrique** | | | | | |
|---|---|---|---|---|---|
| ***Cluster N°*** | ***# comestible*** | ***# toxique*** | ***Cluster N°*** | ***# comestible*** | ***# toxique*** |
| 1 | 666 | 478 | 11 | 120 | 144 |
| 2 | 283 | 318 | 12 | 128 | 140 |
| 3 | 201 | 188 | 13 | 144 | 163 |
| 4 | 164 | 227 | 14 | 198 | 163 |
| 5 | 194 | 125 | 15 | 131 | 211 |
| 6 | 207 | 150 | 16 | 201 | 156 |
| 7 | 233 | 238 | 17 | 151 | 140 |
| 8 | 181 | 139 | 18 | 190 | 122 |
| 9 | 135 | 78 | 19 | 175 | 150 |
| 10 | 172 | 217 | 20 | 168 | 206 |

| ROCK | | | | | |
|---|---|---|---|---|---|
| ***Cluster N°*** | ***# comestible*** | ***# toxique*** | ***Cluster N°*** | ***# comestible*** | ***# toxique*** |
| 1 | 96 | 0 | 12 | 48 | 0 |
| 2 | 0 | 256 | 13 | 0 | 288 |
| 3 | 704 | 0 | 14 | 192 | 0 |
| 4 | 96 | 0 | 15 | 32 | 72 |
| 5 | 768 | 0 | 16 | 0 | 1728 |
| 6 | 0 | 192 | 17 | 288 | 0 |
| 7 | 1728 | 0 | 18 | 0 | 8 |
| 8 | 0 | 32 | 19 | 192 | 0 |
| 9 | 0 | 1296 | 20 | 16 | 0 |
| 10 | 0 | 8 | 21 | 0 | 36 |
| 11 | 48 | 0 | | | |
| **Approche Hansen (coloration minimale)** | | | | | |
| ***Cluster N°*** | ***# comestible*** | ***# toxique*** | ***Cluster N°*** | ***# comestible*** | ***# toxique*** |
| 1 | 0 | 36 | 11 | 0 | 246 |
| 2 | 24 | 24 | 12 | 0 | 1296 |
| 3 | 16 | 8 | 13 | 96 | 248 |
| 4 | 192 | 0 | 14 | 0 | 1692 |
| 5 | 224 | 0 | 15 | 1728 | 0 |
| 6 | 24 | 8 | 16 | 512 | 150 |
| 7 | 288 | 0 | 17 | 192 | 50 |
| 8 | 96 | 42 | 18 | 48 | 32 |
| 9 | 48 | 84 | 19 | 48 | 0 |
| 10 | 672 | 0 | | | |
| **Approche par b-coloration** | | | | | |
| ***Cluster N°*** | ***# comestible*** | ***# toxique*** | ***Cluster N°*** | ***# comestible*** | ***# toxique*** |
| 1 | 0 | 36 | 11 | 139 | 0 |
| 2 | 96 | 464 | 12 | 18 | 0 |
| 3 | 695 | 72 | 13 | 0 | 1296 |
| 4 | 768 | 0 | 14 | 224 | 0 |
| 5 | 1510 | 0 | 15 | 0 | 1728 |
| 6 | 220 | 0 | 16 | 48 | 32 |
| 7 | 145 | 0 | 17 | 192 | 0 |
| 8 | 0 | 288 | | | |
| 9 | 144 | 0 | | | |
| 10 | 9 | 0 | | | |

**Tableau 3.12 - Effectif par classe pour la base "Mushroom"**

En outre, le tableau 3.12[4] montre les différences d'effectif des classes de champignons produites par (1) la b-coloration (17 classes), (2) l'algorithme de Hansen (19 classes), (3)

[4] La couleur grise du fond dans le tableau 3.12 représente les classes non pures (*i.e.* qui comprennent à la fois des champignons toxiques et comestibles).

l'approche de classification hiérarchique saut barycentrique (20 classes) et (4) l'algorithme ROCK (21 classes). Ainsi, nous pouvons constater ce qui suit :

- Toutes les classes identifiées par ROCK sont pures (sauf la classe 15) : chaque classe inclut soit des champignons toxiques, soit comestibles.
- Toutes les classes construites par l'algorithme de classification hiérarchique par saut barycentrique sont des classes non pures.
- Les classes produites par l'algorithme de Hansen hormis neuf classes sont des classes pures.
- Les classes produites par l'algorithme de b-coloration hormis trois classes (les classes 2,3 et 16) sont pures.

***Comparaison des résultats***

Premièrement, il convient de noter, que la qualité de la classification produite par l'algorithme de *classification hiérarchique par saut barycentrique* n'est pas satisfaisante. Chaque classe de cette partition contient un nombre considérable de champignons à la fois toxiques et comestibles, et la taille de ces classes est uniforme. En effet, plus de 90% des classes ont une taille entre 200 et 400, et seulement une classe contient plus de 1000 champignons. Ces résultats sont dus au problème de "représentant unique" dont elles souffrent les approches basées sur le calcul des centres de gravités (section 2.4.2.3). De ce fait, l'algorithme *saut barycentrique* ne peut pas facilement capturer les classes de tailles variées et nous concluons donc que les classes ne sont pas bien séparées.

Les meilleurs résultats sont donnés par l'algorithme ROCK, un algorithme typique aux données catégorielles et qui est proposé comme un algorithme de classification robuste pour de telles données [Guha *et al.,* 2000]. L'algorithme par *b-coloration*, qui est applicable à tout type de données (dans la mesure où nous pouvons construire une matrice de dissimilarités entre les individus à classer), fournit des résultats approximativement proches de ceux de l'algorithme ROCK et plus meilleurs que ceux de l'approche de Hansen (employant une technique de coloration minimale). Ceci confirme que le concept *dominance* conduit à des classes plus significatives et bien-séparés.

***Note***

Il convient de noter que dans toutes les expériences que nous avons menées jusqu'ici sur les jeux de données de l'UCI, la partition optimale retournée par l'approche de *b-coloration* correspond toujours à la plus grande valeur de l'indice de *Dunn généralisé*. Ceci confirme l'idée que cet indice est reconnu comme une bonne évaluation de classification et comme un indice bien approprié pour fournir un compromis entre la maximisation de la dissimilarité interclasse

### 3.4.4. Jeux de données médicales du PMSI

Dans les hôpitaux français, la classification appelé *Groupe Homogène de Malades* (GHM) a été introduite dans les années 80 avec le nouveau *Programme de Médicalisation des Systèmes d'Information* (PMSI). Les données du PMSI sont associées aux traitements chirurgicaux/médicaux, examens et diagnostiques médicaux, aussi bien qu'aux informations personnelles sur le patient et sur son séjour telles que l'âge, le sexe, l'indice de gravité synthétique, le nombre d'unités médicales visitées, la durée du séjour dans le service de réanimation, etc. Un algorithme déterministe sous la forme d'un arbre de décision régit la classification de chaque séjour hospitalier d'un patient dans un des 800 GHM existants. Chaque groupe est prévu pour être homogène en consommation médicale et ressources, et notamment les séjours hospitaliers d'un même GHM devraient avoir probablement la même durée. Il convient de noter que cette classification en GHM joue un rôle considérable dans l'allocation budgétaire des hôpitaux en France.

Malgré les améliorations successives apportées à la classification en GHM, qui est notamment mise à jour chaque deux années par l'Agence Technique de l'information sur l'Hospitalisation (ATIH), les établissements de soins (publics et privés) ont tendance à réfuter l'adéquation de cette classification à leurs besoins et pointent notamment la forte hétérogénéité des classes obtenues. En effet, la catégorisation des séjours en GHM engendre une asymétrie d'information qui résulte de la diversité des pathologies et des prises en charge dans un même GHM [Quantin *et al.*, 1999].

Dans cette section, l'algorithme de classification par *b-coloration* décrit dans ce chapitre a été proposé pour fournir une alternative à la classification en GHM [Elghazel *et al.*, 2006a]. Le nombre de classes recherchées étant inconnu a priori et aucune importance n'a été accordée à sa valeur (qu'elle soit plus grande/petite que le nombre de GHM existants). L'objectif principal des professionnels de santé est d'avoir une partition fine de l'ensemble de séjours, constituée de groupes homogènes (sur l'aspect médical et économique) et bien séparés. Ils ont trouvé l'intérêt dans notre approche grâce sa prise en compte simultanée des deux aspects médicale et économique, représentés dans les données PMSI, dans la conception de nouveaux groupes de séjours. L'algorithme a été examiné sur un grand échantillon de données extrait d'une base PMSI-2003, fournie par l'Agence Régionale d'Hospitalisation Rhône-Alpes (ARH-RA) et qui recense les données PMSI de tous les établissements de santé de la Région (publics et privés). Cette étude a permis d'établir, d'une manière rétrospective, une nouvelle typologie plus fine des séjours. L'exploitation de différents critères de qualité de classification, a permis également d'analyser l'homogénéité des groupes obtenus d'une part, et d'identifier les problèmes associés aux groupes homogènes de malades et de repérer les GHM les plus hétérogènes, d'autre part.

L'échantillon de données utilisé comporte 2750 séjours hospitaliers du système PMSI. Ils correspondent actuellement à 79 GHM recouvrant ~65% des séjours de la base fournie par l'ARH-RA. Afin de garantir la distribution originale des GHM dans l'échantillon choisi (*i.e.* chaque GHM est proportionnellement représenté dans cet échantillon), une procédure d'échantillonnage par *stratification* a été réalisée en utilisant le logiciel SAS. D'autre part, nous avons utilisé, pour la description des séjours, sept attributs : deux de nature

numérique (à savoir, la durée totale du séjour et le nombre d'actes classant opératoires), un de nature binaire (à savoir, la présence/absence de diagnostiques médicales associés), deux de nature catégorielle (à savoir, le mode de sorite du patient et l'âge du patient, que nous avons quantifié en quatre modalités), et deux attributs symboliques (à savoir, le diagnostic médical principal et les actes médicaux).

Soit $X = \{X_1, X_2, \dots, X_n\}$ l'ensemble des séjours hospitaliers à grouper. Chaque séjour est décrit par un mélange de sept variables classiques et symboliques $Y = \{Y_1, Y_2, \dots, Y_7\}$. De a

Similairement aux expériences précédentes avec les jeux de données classiques de l'UCI, la distance euclidienne a été appliquée pour définir le degré de dissimilarité entre les séjours (équation 2.27). La fonction de comparaison $g_h$ entre deux descriptions quantitatives/qualitatives demeure la même que dans l'équation 2.16/2.18. Dans le cas symbolique (ensemble de valeurs), nous avons nous avons cherché, pour définir l'indice de proximité entre les séjours, à prendre en compte et intégrer les connaissances significatives qui concernent les actes et les diagnostics dans le calcul de $g_h$. En effet, comme précisé auparavant, les domaines d'observations des deux variables *diagnostic principal* et *actes médicaux* sont organisés dans une structure hiérarchique : le catalogue d'actes médicaux CdAM ainsi que la Classification Internationale des Maladies (CIM10) pour les diagnostiques. Ces deux informations sont particulièrement caractéristiques du PMSI, nous pouvons ainsi observer que les actes reflètent l'information sur les coûts (consommation de biens et de services de santé) alors que les diagnostics reflètent la connaissance médicale du séjour.

En outre, afin de coupler les deux aspects *structurel* et *sémantique* du CdAM et de la CIM10 dans le calcul de proximité entre les séjours, nous avons choisi d'utiliser la *distance de Hausdorff* défini dans le chapitre précédent (équation 2.24). Puisque cette distance est très sensible aux éléments extrêmes, nous l'avons modifié pour définir la fonction de dissimilarité suivante :

$$\forall\ X_i, X_j \in X;\ g_h\left(Y_h(X_i), Y_h\left(X_j\right)\right)$$
$$= max\left\{\frac{\sum_{a\in Y_h(X_i)}\left(\underset{b\in Y_h(X_j)}{inf}(d(a,b))\right)}{|Y_h(X_i)|}, \frac{\sum_{a\in Y_h(X_j)}\left(\underset{b\in Y_h(X_i)}{inf}(d(a,b))\right)}{\left|Y_h\left(X_j\right)\right|}\right\} \quad (3.9)$$

où $d(a,b)$ est la distance entre deux actes médicaux (respectivement deux diagnostiques) $a$ et $b$. Elle est définie par le nombre d'arêtes séparant les deux actes (respectivement diagnostiques) au premier sommet de l'arbre hiérarchique du CdAM (respectivement CIM10) contenant $a$ et $b$.

Comme précisé auparavant, l'objectif principal est d'obtenir des classes avec une cohésion maximale (similarité *intraclasse*) et un maximum de séparation entre elles (dissimilarité *interclasse*). Pour évaluer la qualité de la partition obtenue, nous avons cherché à comparer notre approche de *b-coloration* avec *la Classification Ascendante*

*Hiérarchique* (*CAH*) (saut minimal), *l'approche de Hansen* et aussi *la classification GHM actuelle*. Le tableau 3.13 suivant fournit les résultats de classification obtenus sur les données PMSI par ces différentes approches de classification.

| ***Approche de classification*** | ***#classes k*** | ***DB*** | ***Dunn_gen*** | ***Cohésion*** | ***Distinction*** |
|---|---|---|---|---|---|
| Approche par b-coloration | 107 | 1.735 | 0.796 | 0.163 | 0.394 |
| Approche Hansen (coloration minimale) | 101 | 1.804 | 0.772 | 0.128 | 0.381 |
| CAH (Saut minimum) | 108 | 1.924 | 0.766 | 0.045 | 0.572 |
| Classification en GHM actuelle | 79 | 1.821 | 0.616 | 0.149 | 0.360 |

**Tableau 3.13 - Performances de la classification sur la base des séjours PMSI**

### *Comparaison des résultats*

Rappelons que la meilleure partition est celle qui minimise l'indice de *Davies-Bouldin* et qui maximise tous les autres critères (*Dunn généralisé*, *cohésion* et *distinction*). L'analyse du tableau précédent montre que la partition optimale renvoyée par notre approche de classification est celle qui améliore le plus la classification GHM. En effet c'est la partition qui apporte le plus en *cohésion*, en *Davies-Bouldin* et en *Dunn généralisé*. Ainsi, les résultats obtenus sont très encourageants et montrent l'intérêt de la méthode pour identifier des groupes homogènes distincts de séjours. Néanmoins, l'approche de *classification hiérarchique par saut minimum* produit des meilleurs résultats que la b-*coloration* selon le critère de distinction, bien qu'elle échoue pour les critères de *cohésion*, *Davies-Bouldin* et de *Dunn généralisé*. Ceci est dû à la mauvaise construction des classes. En effet, nous avons observé, en analysant les effectifs des classes produites par cette approche, que la taille des classes change considérablement (une classe parmi les 108 produites contient 1900 séjours hospitaliers, alors que plus de 50% des classes de la partition sont des classes singleton).

En outre, sur les figures 3.10, 3.11 et 3.12 nous visualisons la répartition des 2750 séjours (*tableau croisé* ou *matrice bloc diagonale*[5]) entre les 79 GHM et respectivement les 108, 101 et 107 classes de la typologie renvoyée par l'application respective de la méthode hiérarchique, l'approche de Hansen et l'approche de b-coloration. Ces matrices fournissent une représentation visuelle de la qualité des groupes de séjours hospitaliers identifiés par ces approches. En effet, les cases blanches (appelée aussi *points exceptionnels* ou *bruit*) situées à l'extérieur des blocs diagonaux donnent des informations sur *l'homogénéité* et la *séparation* des classes de la partition obtenue. Plus nous avons de points exceptionnels dans la matrice, plus les classes contiennent un peu de tout les GHM et donc une diversité de séjours hospitaliers (les classes sont *non significatives* et *mal-séparées*), et vice-versa.

[5] Les matrices bloc diagonale ont été construites par l'algorithme ROC (Rank Order Clustering) (King J. R. *Machine-component grouping in production flow analysis: an approach using rank order clustering algorithm*, International Journal of Production Research, **18**(2), pp. 213-232, 1980.)

Une première analyse des figures 3.10, 3.11 et 3.12 montre qu'un grand nombre des 79 GHM sont subdivisés en en au moins deux classes. Ceci vient confirmer le problème d'hétérogénéité des GHM (*i.e.* une diversité des pathologies et des prises en charge dans un même GHM). Comme nous l'avons mentionné ci-dessus, la matrice bloc diagonale de la classification ascendante hiérarchique par saut minimum (voir figure 3.10) inclut un groupe avec une variété de séjours hospitaliers (plusieurs GHMs sont représentés dans ce groupe contenant plus de 1900 séjours) : ceci est un groupe non homogène. D'autre part, La figure 3.12 indique que la matrice bloc diagonale de la *b-coloration* fournit moins d'éléments exceptionnels que la matrice bloc diagonale de l'algorithme de Hansen (voir figure 3.11).

Cette partie a été validée par des médecins et des spécialistes de l'Agence Régionale d'Hospitalisation Rhône-Alpes (ARH-RA) et de l'Agence Technique de l'Information sur l'Hospitalisation (ATIH), avec qui notamment nous avons validé le choix des attributs pour décrire les individus. Ils ont également mentionné que la classification plus fine fournie par la méthode leur semblait plus réaliste. Ils nous ont ainsi transmis que l'éclatement des GHM en différentes classes correspond d'une part au ressenti du terrain, et que cela rejoignait les modifications qu'ils étaient en train de planifier eux-mêmes par ailleurs dans la nouvelle version des GHM.

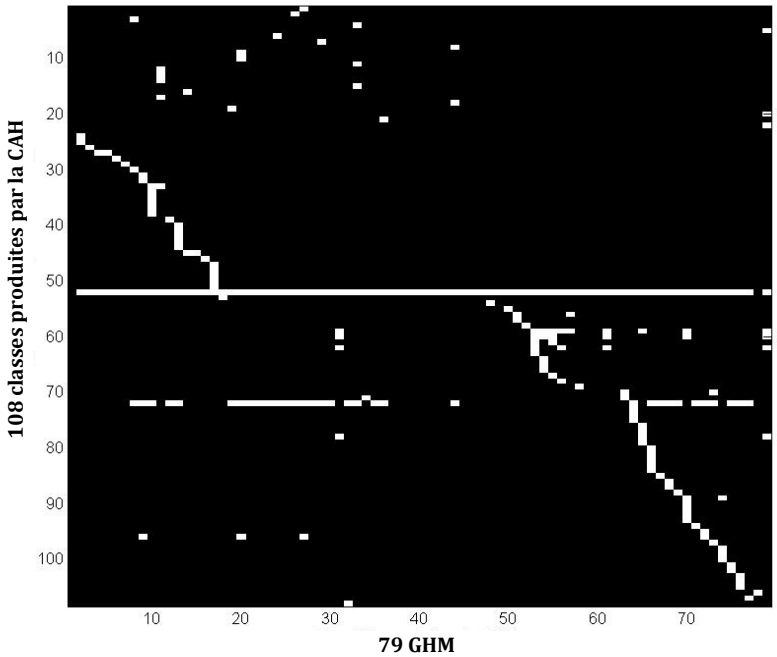



**Figure 3.10 - Matrice bloc diagonale GHM (79) x Classes produites par la CAH (108)**

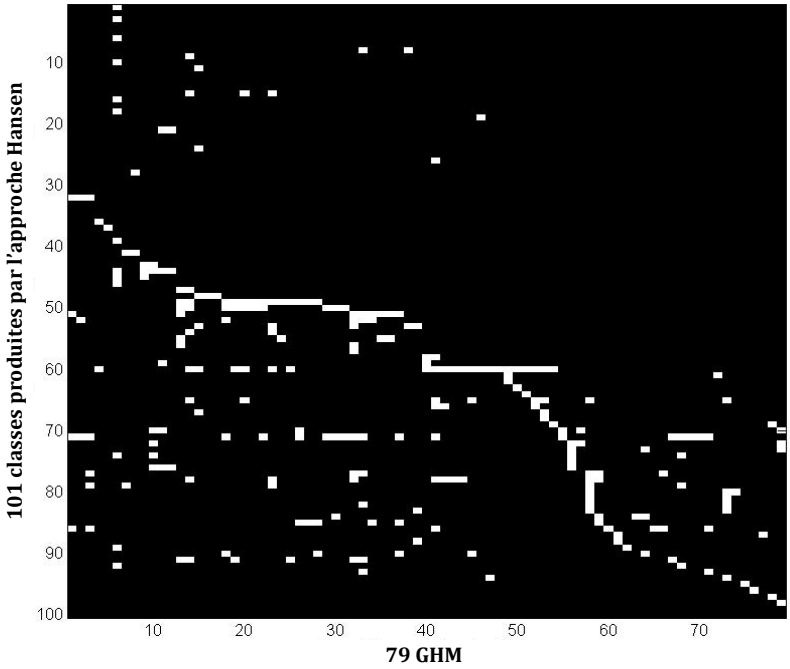



**Figure 3.11- Matrice bloc diagonale GHM (79) x Classes produites par l'approche Hansen (101)**

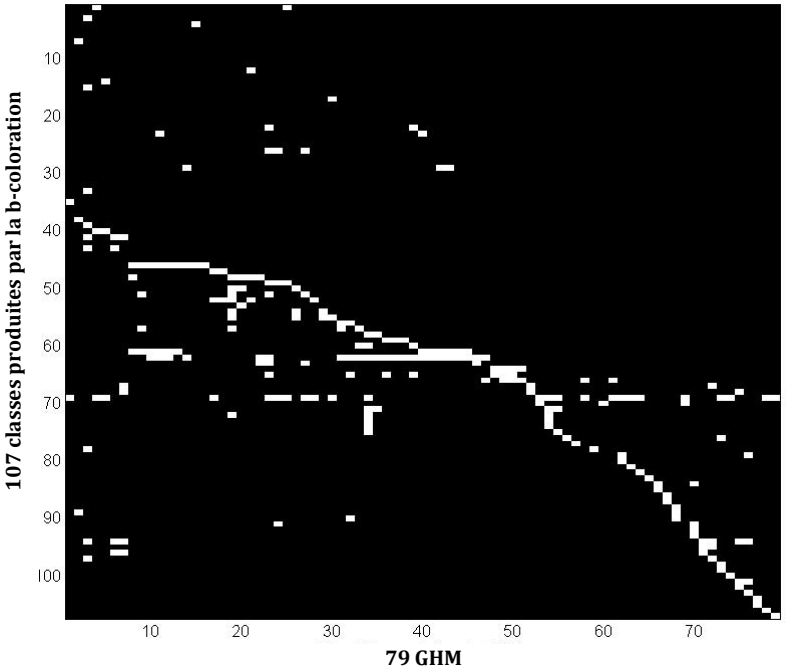



**Figure 3.12- Matrice bloc diagonale GHM (79) x Classes produites par la b-coloration (107)**

### 3.4.5. Jeu d'images archéologiques

Nous avons cherché à vérifier la validité des résultats précédents en utilisant un autre type de données : les *images*. Dans cette optique, nous avons réalisé des expérimentations de l'approche de classification sur un échantillon d'images archéologiques, dans le cadre

du projet européen **TArcHNA**[6] « **T**owards **Arc**heological **H**eritage **N**ew **A**ccessibility ». Ce dernier à pour objectif de donner une meilleure visibilité du patrimoine archéologique aux archéologues, en développant de nouveaux modèles et outils pour l'accès à ce patrimoine, reliant ainsi le passé à l'avenir.

L'échantillon considéré comporte 166 objets archéologiques qui ont été annotés (par les archéologues) selon des informations concernant la forme, la matière première et la période (*c.f.* figure 3.13). Notre objectif est d'établir une classification automatique de ces images de manière à regrouper entre eux les images ayant des thématiques sémantiquement proches.

Les images étant annotées, une matrice d'occurrences *Termes x Images* est construite, dont chaque colonne est associée à une image et contient la présence/absence (1/0) des termes pour l'image concernée. Cette forme de représentation ne traduit pas la proximité sémantique entre les images et souffre des problèmes de *synonymie* et de *polysémie*. Afin d'assurer un rapprochement sémantique entre les vecteurs descripteurs d'images au travers des mots les composant et avant de lancer la procédure de classification automatique, nous avons utilisé une technique d'analyse statistique baptisée *l'indexation sémantique latente* (*Latent Samantic Indexing LSI*) [Deerwester *et al.*, 1990]. L'approche de la LSI consiste à remplacer la représentation *Termes x Images* par une autre traduisant mieux la proximité sémantique entre mots et images en utilisant les mêmes données. Cette approche consiste à réduire les dimensions de la matrice d'occurrences par le biais d'une décomposition en valeurs singulières (*Singular Value Decomposition : SVD*) qui permet de ne garder que les $m$ termes les plus significatifs. Afin de calculer la proximité entre deux images $X_i$ et $X_j$, celles-ci sont représentées dans un espace vectoriel de dimensions $m$. Par la suite, la LSI s'appuie sur une mesure basée sur le calcul du cosinus de l'angle entre les deux nouveaux vecteurs de $X_i$ et $X_j$ pour calculer la distance entre eux : $\left(d(X_i, X_j) = 1 - cos(X_i, X_j)\right)$.

**Figure 3.13 – Base d'images archéologiques**

[6] http://www.tarchna.org/home.htm

Les expérimentations menées sur 166 images archéologiques, ont donné d'excellents résultats, qui ont démontré le bon niveau de performance de notre approche de classification automatique par b-coloration vis-à-vis de la technique des *k-moyennes*. L'analyse des *mots-clés dominants* dans les classes de la typologie obtenue (les mots qui figurent dans l'annotation d'au moins 60% des images de la classe) a été utilisée pour la validation. En effet, contrairement à la technique des *k-moyennes,* où 2 des 4 classes obtenues ne présentent aucun mot-clé dominant (*c.f.* tableau 3.14), l'approche de *classification par b-coloration* fournit un regroupement des 166 images en 8 classes sémantiquement disjointes, où les images d'une même classe traduisent des thématiques sémantiquement proches et effectivement différentes des autres classes d'images (*c.f.* tableau 3.15).

Ajoutons que notre approche de classification a été intégrée dans une plateforme logicielle appelée **KSyDoC** (A **K**eyword-based **S**ystem for **Do**cument **C**lustering and Retrieval) [Azzaoui *et al.*, 2007]. Il s'agit d'un système dédié à la classification et la recherche à base de mots clés dans les banques de documents textuels et images, notamment archéologiques. Ce travail avait été effectué dans le cadre d'un stage de Master 1. Ce qui nous a permis de rapidement appliquer notre approche de classification à ce jeu de données images.

| ***Cluster N°*** | ***Effectif*** | ***Mots clés dominants*** |
|---|---|---|
| 1 | 30 | Orientalizing Period; Pottery |
| 2 | 74 | Orientalizing Period; Pottery; Achromatic; Impasto; Ware |
| 3 | 29 | Pas de mots clés dominants |
| 4 | 33 | Pas de mots clés dominants |

**Tableau 3.14 - Effectif et mots clés dominants par classe d'images pour les k-moyennes**

| ***Cluster N°*** | ***Effectif*** | ***Mots clés dominants*** |
|---|---|---|
| 1 | 76 | Orientalizing Period; Pottery; Achromatic; Impasto; Ware |
| 2 | 7 | Hellenistic Period; Lid; Nenfro; Sarcophagus |
| 3 | 30 | Archaic Period; Pottery; Figure; Red |
| 4 | 6 | Orientalizing Period; Bronze |
| 5 | 9 | Hellenistic Period; Bronze; Mirror |
| 6 | 30 | Scarab; Ornamenta |
| 7 | 7 | Villanovan Period; Bronze; Fibula |
| 8 | 1 | Hellenistic Period; Bronze ; Figure ; Vase |

**Tableau 3.15 - Effectif et mots clés dominants par classe d'images pour la b-coloration**

## 3.5. Algorithme incrémental de classification par b-coloration

### 3.5.1. Introduction

La partition optimale par l'algorithme de classification basé sur la *b-coloration de graphes* étant obtenue (*i.e.* elle correspond au seuil optimal $\theta_o$), le problème maintenant est de savoir comment gérer un flux de données entrant (séjours hospitaliers dans notre cadre d'application) de sorte à affecter une classe aux nouvelles données et effectuer ainsi la mise à jour de la partition optimale obtenue, ou même un flux de données sortant de sorte à réarranger la partition quand certaines données existantes sont retirées de la population.

Considérons la représentation topologique de l'ensemble d'individus $X = \{X_1, X_2, \dots, X_n\}$ regroupés dans une partition $P = \{C_1, C_2, \dots, C_k\}$, par le graphe seuil supérieur optimal $G = (\boldsymbol{V}, \boldsymbol{E})$. L'introduction d'un $n+1^{ème}$ individu $X_{n+1}$ dans le système ($n$ individus étant préalablement classés) correspond à l'ajout d'un sommet $v_{n+1}$ au graphe seuil optimal ($i.e.\ \boldsymbol{V} = \boldsymbol{V} \cup \{v_{n+1}\}$). Cet ajout s'accompagne d'ajout d'arêtes entre ce sommet et les autres sommets dont la dissimilarité avec $v_{n+1}$ est supérieure à $\theta_o$ ($i.e.\ \boldsymbol{E} = \boldsymbol{E} \cup \{(v_i, v_{n+1}) | v_i \in \boldsymbol{V}$ et $d(X_i, X_{n+1}) > \theta_o\}$). Cependant, le retrait d'un individu $X_m$ de l'ensemble total des données $X$, correspond à la suppression de son sommet relative $v_m$ du graphe seuil optimal ($i.e.\ \boldsymbol{V} = \boldsymbol{V} - \{v_m\}$), ainsi que des arêtes dont il fait parti ($i.e.\ \boldsymbol{E} = \boldsymbol{E} - \{(v_i, v_m) | v_i \in \boldsymbol{V}$ et $d(X_i, X_m) > \theta_o\}$).

Dans la suite de ce chapitre, nous proposons un algorithme incrémental de classification non supervisée [Elghazel *et al.*, 2007c,d]. Cet algorithme exploite d'une part la connaissance des dissimilarités entre chaque paire d'individus et d'autre part la propriété de dominance de la *b-coloration*. De plus, l'algorithme respecte l'ensemble des contraintes suivantes :

- L'affectation d'une *couleur* (*classe*) au nouveau sommet $v_{n+1}$ ou le réarrangement de couleurs lors du retrait d'un sommet $v_m$ doit maintenir la qualité de la nouvelle partition d'une part (en terme de l'indice de validité de *Dunn généralisé*[7]), et les propriétés de la *b-coloration* d'autre part (*coloration propre* et *dominance*).
- La qualité de la partition générée par l'algorithme incrémental doit être équivalente à la qualité de la partition qui serait fournie par le système si tout le processus de *b-coloration* était relancé. Enfin l'ordre de complexité devra être inférieur à celui correspondant à la relance du processus de *b-coloration*.

Supposons que les sommets de $G$ sont colorés, les notations supplémentaires suivantes seront considérées dans la suite de ce chapitre :

[7] Nous avons choisi d'utiliser l'indice de Dunn généralisé vu ses performances dans les expérimentations menées dans les sections précédentes.

- $Dom(v_i)$ : la dominance du sommet $v_i$. $Dom(v_i) = 1$ si $v_i$ est un sommet dominant de sa couleur $c(v_i)$ et $Dom(v_i) = 0$ sinon.
- $k$ : le nombre actuel de couleurs (classes) dans $G$.

## 3.5.2. Ajout d'une instance

Comme précisé auparavant, l'algorithme incrémental se base d'une part sur la connaissance des dissimilarités entre les objets et d'autre part sur la propriété de dominance de la *b-coloration*. Lorsqu'un nouveau sommet $v_{n+1}$ est introduit dans le graphe $G$, trois cas sont possibles :

- le nouveau sommet s'intègre à l'une des $k$ couleurs existantes ;
- nous créons une nouvelle couleur pour le nouveau sommet ;
- l'arrivée du nouveau sommet entraîne la suppression et la réaffectation de certaines couleurs existantes.

C'est la notion de dominance des clusters qui va permettre de formaliser les différentes situations possibles. Ainsi, à l'arrivée d'un sommet $v_{n+1}$ dans le graphe $G$, nous sommes en présence des scénarios suivants :

### *3.5.2.1. Scénario 1 : Au moins un voisin dominant à $v_{n+1}$ par couleur*

Le sommet $v_{n+1}$ a au moins un voisin dominant dans chacune des $k$ couleurs (*i.e.* la dissimilarité entre $v_{n+1}$ et ce dominant est supérieure à $\theta_o$) comme le définit l'expression suivante :

$$\forall\ C_h \in P = \{C_1, C_2, \dots, C_k\};\ \exists v \in \left(C_h \cap N(v_{n+1})\right)\ tel\ que\ Dom(v) = 1 \qquad (3.10)$$

Dans ce cas, une nouvelle couleur (une $k+1^{ème}$) est créée pour le sommet $v_{n+1}$ (voir figure 3.14). Dans le cas contraire, le scénario 2 est réalisé.

***Proposition 3.7*** *Après l'ajout de la nouvelle classe de couleur, la coloration en k+1 couleurs de G est une b-coloration.*

***Preuve*** $\forall\ C_h \in P = \{C_1, C_2, \dots, C_k\}\ \exists\ v \in \left(C_h \cap N(v_{n+1})\right)$ *tel que* $Dom(v) = 1$. *Ainsi,* $Dom(v_{n+1}) = 1$ *dans sa couleur (la couleur k+1) et le sommet v est toujours dominant de sa couleur vu qu'il était un dominant et qu'il a un voisin* $v_{n+1}$ *dans la nouvelle couleur k+1* $(i.e. Dom(v) = 1)$. *En conséquence,* $\forall\ C_h \in P = \{C_1, C_2, \dots, C_k, C_{k+1}\}\ \exists\ v \in C_h$ *tel que* $Dom(v) = 1$ *: la coloration en k+1 couleurs du graphe G est une b-coloration.*

Comme mentionné, la création de la nouvelle couleur peut conduire à des modifications de couleurs de certains sommets, toujours dans l'objectif d'améliorer la qualité de la partition obtenue. Ceci n'est faisable que si ce changement ne viole aucune contrainte de la *b-coloration* (*coloration propre* et *dominance*). Pour faciliter la compréhension du problème, nous avons défini les définitions suivantes :

***Définition 3.1*** *Un sommet $v_s$ est dit "sommet support" s'il est le seul sommet coloré avec $c(v_s)$ dans le voisinage $N(v_d)$ d'un sommet dominant $v_d$ d'une autre couleur. Ce sommet $v_s$ ne peut pas changer de couleur (sous réserve de casser la dominance du voisin précité) sauf si sa couleur est supprimée du graphe.*

***Définition 3.2*** *Un sommet $v_c$ est dit "sommet critique" s'il est un dominant de sa couleur $(i.e.\ Dom(v_c) = 1)$ ou un sommet support. Ainsi, $v_c$ ne peut pas être recoloré.*

***Définition 3.3*** *Un sommet $v$ est dit "sommet non voisin d'une couleur C" si C n'est pas dans le voisinage couleur de $v$ $\left(i.e.\ C \notin N_c(v)\right)$.*

***Définition 3.4*** *Un sommet $v$ est dit "sommet libre par rapport à une couleur C" s'il est non critique et qu'il est non voisin de C. Ainsi, la couleur C peut être assignée à $v$.*

Pour évaluer la pertinence (amélioration de la qualité de partitionnement) du passage à la couleur $C$ d'un *sommet libre* $v$ par rapport à cette couleur $C$, nous calculons la dissimilarité entre le sommet $v$ et la couleur $C$ définie comme la dissimilarité moyenne entre l'individu $x_v$ associé à $v$ et les autres individus de la couleur $C$ (équation 3.11). Si cette mesure est inférieure à la dissimilarité moyenne entre $v$ et sa couleur actuelle, la couleur de $v$ est modifiée, sinon il garde sa couleur.

$$dist(v, C) = \frac{1}{|C|}\sum_{y \in C} d(x_v\ , y) \tag{3.11}$$

La minimisation d'une telle distance est liée à la minimisation de la *dissimilarité intraclasse*, ce qui peut améliorer dans certains cas la qualité de la partition en augmentant la valeur de l'indice de *Dunn généralisé*.

***Définition 3.5*** *Supposons que le sommet $v$ (associé à l'individu $x_v$) initialement coloré avec $c(v)$ est recoloré avec $C$. Ainsi, cette transformation induit des changements dans les dissimilarités $dist(v_i, C)$ et $dist(v_i, c(v))$ pour tout sommet $v_i$ de $G$. Par ailleurs, le calcul de ces dissimilarités est de l'ordre de $O(n^2)$, car il s'effectue pour tous les couples (sommet, couleur) du graphe. Le nombre de ces opérations peut être réduit à $O(n)$ si nous gardons l'historique de ces dissimilarités. En effet les nouvelles dissimilarités sont obtenues à partir des anciennes par les équations de mise à jour suivantes :*

- Rattachement du sommet $v$, associé à l'individu $x_v$, à la couleur $C$ :

$$dist^{nouv}(v_i, C) = \frac{|C| dist^{anc}(v_i, C) + d(X_i\ , x_v)}{|C| + 1} \tag{3.12}$$

- Changement de la couleur $c(v)$ du sommet $v$, associé à l'individu $x_v$ :

$$dist^{nouv}(v_i, c(v)) = \frac{|c(v)| dist^{anc}(v_i, c(v)) - d(X_i\ , x_v)}{|c(v)| - 1} \tag{3.13}$$

***Entrée***: *Insertion d'un sommet $v_{n+1}$ ayant au moins un voisin dominant par couleur*
***Sortie***: *Une $b-$ coloration de $G$*

1: $c(v_{n+1}) \leftarrow k+1$;
2: ***pour tout*** *sommet $v_i$ libre par rapport à la couleur $k+1$* ***faire***
3: ***si*** $dist(v_i, k+1) < dist(v_i, c(v_i))$ ***alors***
4: ***pour tout*** *sommet $v_j$ de $G$* ***faire***
5: ***Mise_à_jour*** $\left(dist(v_j, k+1)\right)$;//éq. 3.12, *rattachement de $v_i$ à $k+1$*
6: ***Mise_à_jour*** $\left(dist\left(v_j, c(v_i)\right)\right)$;//éq. 3.13, *changement de $c(v_i)$*
7: ***fin pour***
8: $c(v_i) \leftarrow k+1$;
9: ***fin si***
10: ***fin pour***
11: ***Recherche_dominant***();

**Procédure** Scénario_1 ()

Les couleurs de certains sommets de $G$ étant changées, la procédure *Recherche_dominant()* -de complexité *O(n)*- permet l'identification des nouveaux sommets dominants dans *G*.

***Proposition 3.8*** *La complexité de la procédure du scénario 1 est de l'ordre de O(n²).*

***Preuve*** *Une nouvelle couleur étant créée pour $v_{n+1}$. La procédure du scénario 1 teste pour chaque sommet libre vers cette nouvelle couleur (au maximum n) s'il peut l'intégrer. Si c'est le cas une mise à jour de la valeur de dissimilarité entre chaque sommet du graphe et cette nouvelle couleur est faite en O(n). Ainsi, la procédure du scénario 1 utilise au maximum (n*n) instructions : la complexité est de O(n²).*

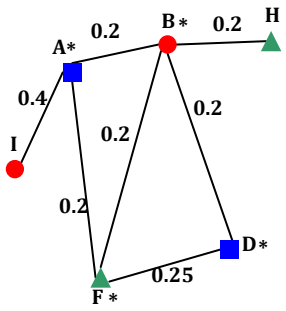



**Figure 3.14 - Partition en 3 classes des sommets A,B,D,F,H et I (θ=0.15)**

"*" indique la dominance d'un sommet

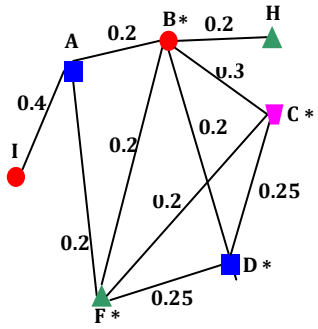



**Figure 3.15 - Mise à jour de la partition de la figure précédente après l'ajout du sommet C**

Ajout d'une nouvelle couleur. **C** est adjacent à au moins un sommet dans chacune des 3 couleurs

### *3.5.2.2. Scénario 2 : Aucun voisin dominant à $v_{n+1}$ dans m couleurs*

Le nouveau sommet $v_{n+1}$ n'a aucun voisin (*i.e.* aucune dissemblance) parmi tous les dominants de *m* couleurs (*1≤m≤k*). Ces couleurs sont ainsi "*prêtes à accueillir le nouveau sommet $v_{n+1}$* ". Deux cas se présentent alors :

### a. Scénario 2.1 : Aucun voisin à $v_{n+1}$ dans $m_1 \leq m$ couleurs

Si $v_{n+1}$ est un *non voisin* de $m_1$ parmi les $m$ couleurs, il intègre l'une des $m_1$ couleurs, celle avec laquelle il a la plus faible dissimilarité. Une telle couleur est dite *couleur gagnante* notée $C$ (voir figure 3.16). Dans le cas contraire nous réalisons le *scénario 2.2*.

***Entrée***: *Insertion d'un sommet $v_{n+1}$ qui n'a aucun voisin dans $m_1$ couleurs*
***Sortie***: *Une $b-$coloration de $G$*
1: $H \leftarrow \{h |\ h \notin N_c(v_{n+1})\}$;// *un ensemble des $m_1$ couleurs*
2: $C \leftarrow argmin_{h \in H}\big(dist(v_{n+1}, h)\big)$;
3: $c(v_{n+1}) \leftarrow C$;
4: ***pour tout*** *sommet $v_i$ de $G$* ***faire***
5: ***Mise_à_jour*** $\Big(dist\big(v_i, c(v_{n+1})\big)\Big)$;// *éq.* 3.12, *rattachement de $v_{n+1}$ à $C$*
6: ***fin pour***
7: ***pour tout*** *sommet $v_i \in N(v_{n+1})$* ***faire***
8: ***test_dominance***$(v_i)$;
9: ***fin pour***

**Procédure** Scénario_2.1 ()

Après l'insertion du sommet $v_{n+1}$ dans la couleur gagnante (parmi les $m_1$ couleurs), certains sommets voisins de $v_{n+1}$ deviennent dominants. Ce sont les sommets à qui manquait exactement un voisin de la couleur gagnante pour être dominant. L'affectation de $v_{n+1}$ à la couleur peut donc modifier la répartition des dominants (apparition de nouveaux dominants). Ce examen est effectué par la procédure *test_dominance(sommet)* de complexité *O(1)*.

***Proposition 3.9*** *La complexité de la procédure du scénario 2.1 est de l'ordre de O(n).*

***Preuve*** *Le nouveau sommet $v_{n+1}$ intègre la couleur C choisie par l'algorithme. Une mise à jour de la valeur de dissimilarité entre chaque sommet du graphe et la couleur C est faite en O(n). Ensuite pour tous sommets voisins de $v_{n+1}$ (au maximum Δ) un test de dominance est fait en O(1). Ainsi, la procédure du scénario 2.1 utilise au maximum (n+Δ*1) instructions : la complexité est de O(n).*

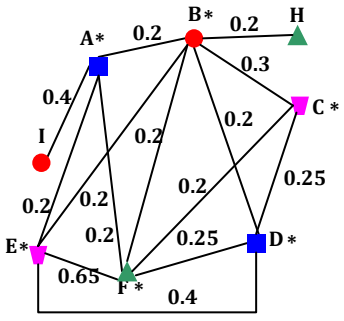



**Figure 3.16 - Mise à jour de la partition de la figure précédente après l'ajout du sommet E**

**E** est non voisin d'une seule couleur (la couleur du sommet **C**)

### b. Scénario 2.2 : Au moins un voisin à $v_{n+1}$ dans les m couleurs

C'est le cas où $v_{n+1}$ ne peut intégrer aucune des $m$ couleurs. Il s'agit d'un sommet $(i.e.\, v_{n+1})$ ayant au moins un voisin non dominant dans chacune des $m$ couleurs :

$$\forall\ C_h\ une\ des\ m\ couleur\ \ \exists\ v \in \left(C_h \cap N(v_{n+1})\right)\ tel\ que\ Dom(v) = 0 \qquad (3.14)$$

Dans ce cas de figure il convient de distinguer à nouveau deux sous-scénarios complémentaires suivants:

- ***Scénario 2.2.1 : Nombre de couleurs m=1***

Si *m=1* (une seule couleur notée $C$ est prête à accueillir $v_{n+1}$), le sommet prend cette couleur et cette dernière subit quelques transformations. En effet, pour conserver une coloration *propre et dominante*, tous les autres sommets de $C$ voisins à $v_{n+1}$ sont rattachés à d'autres couleurs. Si nous avons le choix entre plusieurs couleurs pour un sommet donné, la couleur possédant la plus faible dissimilarité avec ce sommet (équation 3.11) sera choisie pour un bon rattachement.

***Proposition 3.10*** *A l'issue des transformations effectuées sur la couleur C, la nouvelle configuration en k couleurs de G est une b-coloration.*

***Preuve*** $\forall\ v$ *un sommet du graphe tel que* $c(v) = C$ *et* $v \in N(v_{n+1})$ *nous avons* $Dom(v) = 0$. *D'après la propriété de dominance de la b-coloration,* $\exists h \in \{1,2,\dots,k\}$ *tel que* $C_h \neq C$ *et* $C_h \notin N_c(v)$. *De ce fait, le sommet* $v$ *peut être rattaché à la couleur* $C_h$ *et nous obtenons ainsi une coloration propre. D'autre part,* $\forall\ h \in \{1,2,\dots,k\}$ *tel que* $C_h \neq C,\ \ \exists\ v \in \left(C_h \cap N(v_{n+1})\right)$ *tel que* $Dom(v) = 1$. *Ainsi* $v$ *est toujours dominant de sa couleur (i.e.* $Dom(v) = 1$*)* $Dom(v_{n+1}) = 1$. *Ceci donne,* $\forall\ C_h \in P = \{C_1, C_2, \dots, C_k\}\ \exists\ v\ tel\ que\ c(v) = C_h\ et\ Dom(v) = 1$ *: la coloration du graphe G est dominante. En conséquence, nous avons une b-coloration.*

***Entrée***: *Insertion de $v_{n+1}$ qui a au moins un voisin non dominant dans une seule couleur C*
***Sortie***: *Une b − coloration de G*

1: $c(v_{n+1}) \leftarrow C$; // *C est la couleur prête à acceuilir* $v_{n+1}$
2: ***pour tout*** *sommet* $v_i$ *de G* ***faire***
3: ***Mise_à_jour*** $\left(dist(v_i, c(v_{n+1}))\right)$;// *éq.* 3.12, *rattachement de* $v_{n+1}$ *à C*
4: ***fin pour***
5: ***pour tout*** *sommet* $v_i \in N(v_{n+1})$ *tel que* $c(v_i) = C$ ***faire***
6: $H \leftarrow \{h |\ h \notin N_c(v_i)\}$;
7: $coul \leftarrow argmin_{h \in H}(dist(v_i, h))$;
8: ***pour tout*** *sommet* $v_j$ *de G* ***faire***
9: ***Mise_à_jour*** $\left(dist(v_j, coul)\right)$;// *éq.* 3.12, *rattachement de* $v_i$ *à coul*
10: ***Mise_à_jour*** $\left(dist\left(v_j, c(v_i)\right)\right)$;// *éq.* 3.13, *changement de* $c(v_i)$
11: ***fin pour***
12: $c(v_i) \leftarrow coul$;
13: ***fin pour***
14: ***pour tout*** *sommet* $v_i \in N(v_{n+1})$ ***faire***
15: ***test_dominance***$(v_i)$;
16: ***fin pour***

**Procédure** Scénario_2.2.1 ()

***Proposition 3.11*** *La complexité de la procédure du scénario 2.2.1 est de l'ordre de O(nΔ).*

***Preuve*** *La couleur C est attribuée au sommet $v_{n+1}$. Tous les sommets voisins de $v_{n+1}$ coloré par C (au maximum Δ) changeront de couleur. Suite à cette transformation de coloration de G, une mise à jour de la valeur de dissimilarité entre chaque sommet du graphe et la couleur C est faite en O(n). Ensuite pour tous sommets voisins de $v_{n+1}$ (au maximum Δ) un test de dominance est fait en O(1). Ainsi, la procédure du scénario 2.2.1 utilise au maximum (Δ*n+Δ*1) instructions : la complexité est de O(nΔ).*

- ***Scénario 2.2.2 : Nombre de couleurs m>1***

Si *m>1* (plusieurs couleurs sont prêtes à accueillir $v_{n+1}$). Dans ce cas, nous avons besoin de définir la propriété de transformation de couleur suivante :

***Définition 3.5*** *Une couleur C parmi les m couleurs prêtes à accueillir $v_{n+1}$ est dite "objet de transformation" si sa transformation ne viole pas les contraintes de b-coloration pour les autres (m-1) couleurs. En d'autres termes, une couleur C (parmi les m couleurs en question) n'est pas objet de transformation s'il existe au moins une couleur C' (parmi les m) telle que tous les voisins des dominants de C' dans C sont voisins (dissemblables) de $v_{n+1}$.*

***Exemple :*** A tire d'illustration, la figure 3.17 présente deux couleurs $C_1$ et $C_2$ prêtes à accueillir un sommet **F** (*m=2*). L'unique voisin dans la couleur $C_1$ du dominant de la couleur $C_2$ (le sommet **B**) est le sommet **A** (appelé sommet support) qui est adjacent à **F**. Ainsi, la couleur $C_1$ n'est pas *objet de transformation*. En effet, si **F** intègre la couleur $C_1$, le sommet **A** (voisin de **F**) devrait être affecté à une autre couleur et le seul dominant de $C_2$ (**B**) se retrouve sans aucun voisin (sommet dissemblable) dans la couleur $C_1$. En conséquence la transformation de la couleur $C_1$ est interdite : $C_1$ n'est pas prête à accueillir

**F**. Contrairement à $C_1$, la couleur $C_2$ est *objet de transformation* puisque l'un des voisins du dominant de la couleur $C_1$ (le sommet **C**) dans la classe $C_2$ (**E**) n'est pas voisins de **F**. Ainsi, $C_2$ peut accueillir **F**.

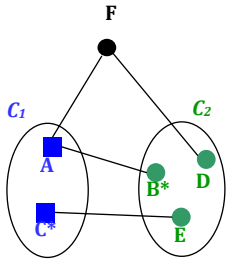



**Figure 3.17 - Identification des couleurs objet de transformation**

Nous avons donc vu qu'une couleur peut subir des transformations à l'issue de l'intégration d'un nouveau sommet (exclusion de sommets, changement de dominant). Seules les couleurs ne détruisant pas les propriétés de la *b-coloration* sont transformables, c'est pourquoi une étape de l'approche incrémentale va consister à évaluer quelles sont les $m_2$ couleurs (parmi $m$) qui font l'*objet de transformation.* Les cas de figure suivants sont alors envisagés :

- *Scénario 2.2.2.1 : Une couleur est objet de transformation*

Si une seule couleur $C$ $(m_2=1)$ parmi les $m$ couleurs est identifiée comme *objet de transformation*, alors la transformation est autorisée. Il suffit ainsi d'exécuter la procédure du scénario 2.2.1 précédent.

- *Scénario 2.2.2.2 : $m_2$>1 couleurs sont objets de transformation*

Si $m_2$ $(1<m_2\leq m)$ parmi les $m$ couleurs sont *objets de transformation*, alors la couleur ayant la plus petite dissimilarité avec $v_{n+1}$ sera choisie pour l'accueillir. Une telle couleur est dite *couleur gagnante*. Sachant que lors de la transformation de cette *couleur gagnante*, les sommets voisins (dissemblables) de $v_{n+1}$ dans cette couleur seront répartis dans d'autres couleurs, la formule employée pour calculer la dissimilarité entre $v_{n+1}$ et les couleurs "*objet de transformation*" doit tenir compte que des sommets non voisin (semblables) à $v_{n+1}$ au sein de ces couleurs. Une fois la *couleur gagnante* choisie, la procédure du scénario 2.2.1 précédent est exécutée.

- *Scénario 2.2.2.3 : Aucune couleur n'est objet de transformation*

Si aucune des $m$ couleurs $(m_2=0)$ n'est *objet de transformation*, une nouvelle couleur *"k+1"* est créé pour $v_{n+1}$ qui sera alors son dominant $(i.e. Dom(v_{n+1}) = 1)$. Suite à cette transformation, les $m$ couleurs serons sans aucun sommet dominant. En effet, leurs dominants n'ont aucun voisinage avec la nouvelle couleur (ne sont pas dissemblables de la $k+1^{ème}$) et deviennent ainsi des sommets non dominants.

Face ce problème, la procédure *Réarranger_b-coloration()* suivante cherche une *b-coloration* du graphe $G$ où toutes les couleurs sont dominantes. Cette procédure s'applique

sur chaque couleur non dominante (*i.e.* les $m$ couleurs) et tente de la modifier après sa suppression du graphe $G$. En effet, étant donné une couleur non dominante $C$, pour chaque sommet $v_i$ coloré par $C$ $(i.e.\ c(v_i) = C)$, la procédure suivante sélectionne une nouvelle couleur pour $v_i$ qui soit différente des couleurs de ses sommets voisins. Dans le cas où nous avons le choix entre plusieurs couleurs pour $v_i$, l'algorithme choisit la couleur dont la dissimilarité avec $v_i$ est la minimale. Dans la suite, nous avons besoin de définir les notations supplémentaires suivantes :

- $L$ : l'ensemble des $k+1$ couleurs utilisées pour la coloration de graphe $G$.
- $D_m$ : l'ensemble des *couleurs dominantes* (*i.e.* avec au moins un sommet dominant). Il comporte initialement toutes les couleurs de $G$ hormis les $m$ couleurs *non objet de transformation.*
- $ND_m$ : l'ensemble des couleurs *non dominantes*. Il comporte initialement les $m$ couleurs *non objet de transformation* $(i.e.\ ND_m = L \setminus D_m)$.

***Entrée***: *Le graphe G contient m couleurs non dominantes*
***Sortie***: *Une b − coloration de G*

1: ***répéter***
2: $C \leftarrow max\{c | c \in ND_m\}$;
3: $L \leftarrow L \setminus \{C\}$;
4: $ND_m \leftarrow L \setminus D_m$;
5: ***pour tout*** *sommet* $v_i$ *tel que* $c(v_i) = C$ ***faire***
6: $H \leftarrow \{h | h \in L \text{ et } h \notin N_c(v_i)\}$;
7: $c(v_i) \leftarrow argmin_{h \in H}\big(dist(v_i, h)\big)$;
8: ***pour tout*** *sommet* $v_j$ *de* $G$ ***faire***
9: ***Mise_à_jour***$\Big(dist\Big(v_j, c(v_i)\Big)\Big)$;// éq. 3.12, *rattachement de* $v_i$ à $c(v_i)$
10: ***fin pour***
11: ***fin pour***
12: ***pour tout*** *sommet* $v_i$ *tel que* $c(v_i) \in ND_m$ ***faire***
13: ***Mise_à_jour***$\big(N_c(v_i)\big)$;
14: ***si test_dominance***$(v_i)$ ***alors***
15: ***Ajout***$(c(v_i), D_m)$;// $c(v_i)$*devient une couleur dominante*
16: ***fin si***
17: ***fin pour***
18: ***jusqu'a*** $(ND_m = \emptyset)$

**Procédure** Réarranger_b-coloration

***Proposition 3.12*** *La procédure du scénario 2.2.2.3 génère une b-coloration de G en O(n²m) où m est le nombre de couleurs non dominantes.*

***Preuve*** *La procédure du scénario 2.2.2.3 est appliquée pour chaque couleur C non dominante (au maximum m). La couleur C est supprimée du graphe, chaque sommet $v_i$ coloré avec C (au maximum n) est recoloré et pour chaque sommet $v_j$ (au maximum n), sa dissimilarité avec la nouvelle couleur est mise à jour. Ensuite, pour chaque sommet du graphe dont la couleur est non dominante (au maximum n), son voisinage en couleurs est mis à jour en examinant les couleurs de ses voisins (au maximum Δ) et un test de dominance est effectué pour vérifier si*

*sa couleur est devenue dominante. Ainsi La procédure du scénario 2.2.2.3 utilise au maximum ((n*n+n*Δ)*m) instructions : la complexité est de O($n^2$m).*

### *3.5.2.3. Discussion*

Nous avons envisagé, comme le défini les paragraphes précédents (voir figure 3.18), en plus de l'attribution du nouveau sommet à une couleur existante, de la création d'une nouvelle couleur et de la suppression de certaines couleurs, d'améliorer la qualité de la partition obtenue en modifiant la couleur de certains sommets. De telles transformations entraînent un raffinement de cette partition en minimisant éventuellement la *dissimilarité intraclasse* qui est une fonction monotone croissante du seuil de dissimilarité $\theta$. Ceci permettra éventuellement de baisser le seuil de dissimilarité optimal $\theta_o$, au cours du processus incrémental.

En addition à ces modifications, nous avons envisagé de compléter cette mise à jour de la partition obtenue par des opérations de fusion de classes. En effet, la fusion de deux classes est déclenchée, après l'insertion d'un certain nombre de sommets, si nous détectons que leur *dissimilarité interclasse* est inférieure au seuil optimal $\theta_o$. Ceci entraînera une augmentation du seuil optimal $\theta_o$ qui pourra violer les contraintes de dominance de certaines couleurs. Dans ce cas, la procédure *Réarranger_b-coloration* définie précédemment sera appliquée sur chaque couleur non dominante pour chercher une nouvelle *b-coloration* du graphe *G*.

Dans les approches incrémentales de classification, il est toujours préférable que l'échantillon de données (initialement choisi pour obtenir une partition optimale qui est passée en entrée du processus incrémental), soit représentatif de l'ensemble de l'espace des variables afin que cette partition soit robuste en forme et en structure. Les mises à jour ainsi définies ci-dessus permettront également de gérer les échantillons de données aléatoirement choisis.

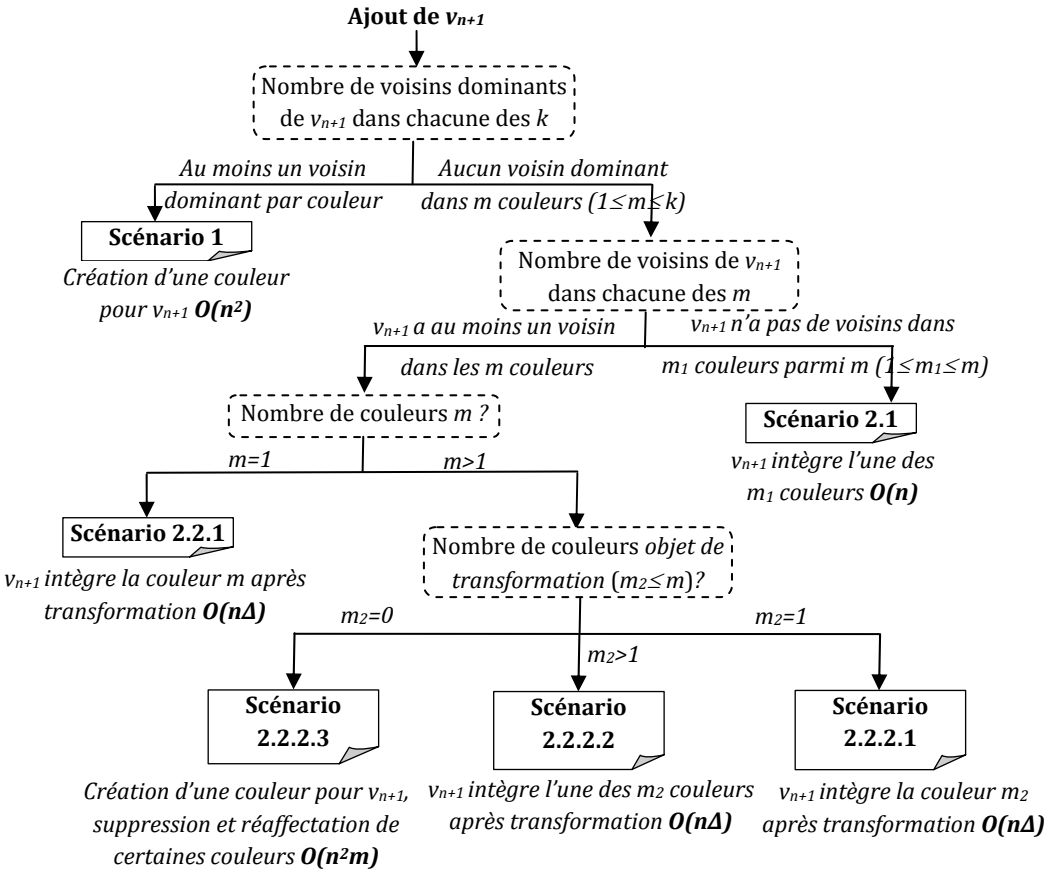



**Figure 3.18 - Arbre de décision pour la mise à jour de la partition en cas d'ajout d'une nouvelle donnée**

## 3.5.3. Suppression d'une instance

Le retrait d'un individu $X_m$ de l'ensemble total des données $X$ correspond à la suppression de son sommet relative $v_m$ du graphe seuil optimal, ainsi que des arêtes dont il fait parti. Ces opérations de suppression affectent la coloration du graphe $G$. Ce dernier requiert ainsi un réarrangement de ses couleurs pour maintenir les propriétés de b-coloration et les bonnes qualités de classification.

D'une manière analogue aux scénarios d'ajout précédents, c'est la notion de dominance des classes qui va permettre de formaliser les différentes situations possibles. Ainsi, lors de la suppression d'un sommet $v$ du graphe $G$, nous sommes en présence des scénarios suivants :

### *3.5.3.1. Scenario 3 : $v_m$ est le seul dominant de sa couleur $c(v_m)$*

Dans ce cas, le sommet $v_m$ est l'unique sommet dominant de sa couleur. Par conséquent, la suppression de $v_m$, affecte la dominance de la couleur $c(v_m)$ qui devient sans sommets dominants et la coloration de $G$ n'est plus une *b-coloration*. En conséquence, les couleurs des sommets restants de $c(v_m)$ doivent être changées. Pour chaque

sommet $v_i$ parmi ces derniers, la transformation est faisable parce qu'il n'est pas dominant. Si nous avons le choix entre plusieurs couleurs pour $v_i$, la couleur possédant la plus faible dissimilarité avec ce sommet (équation 3.11) sera choisie pour un bon rattachement. Dans le cas contraire nous réalisons le *scénario 4.*

#### *3.5.3.2. Scenario 4 : $v_m$ est un sommet support de tous les dominants d'une couleur*

Dans cette situation, le sommet $v_m$ est l'unique sommet coloré par $c(v_m)$ dans le voisinage de tous les sommets dominants d'au moins un couleur $C$. En conséquence, la suppression de $v_m$, pousse ces sommets à devenir non dominants et $C$ sans aucun sommet dominant. Pour résoudre ce problème, la procédure *Réarranger_b-coloration* employée dans le scénario 2.2.2.3 pour trouver une *b-coloration* de $G$ est appliquée pour chaque couleur $C$ non dominante.

Si la suppression du sommet $v_m$ ne vérifie aucun des scénarios 3 et 4 précédents, l'algorithme incrémental supprime le sommet $v_m$ sans aucun réarrangement et la nouvelle coloration de $G$ demeure une *b-coloration*.

### 3.5.4. Expérimentations et performances

Pour évaluer les performances de l'approche incrémentale présentée ci-dessus, nous l'avons implémentée et testée sur trois jeux de données benchmark de l'UCI [Blake and Merz, 1998]. Il s'agit de :

- La base "*Zoo*" : présentée dans 3.4.3.1 comme un jeu de 100 instances d'animaux caractérisées par 17 variables hétérogènes dont 1 qualitative, 1 quantitative et 15 booléennes.
- la base "*Auto*" : présentée dans 3.4.3.4 comme un jeu de 193 instances de voitures caractérisées par 24 variables dont 14 quantitatives et 10 qualitatives.
- la base "*Tic-tac-toe*" : est un jeu de 958 instances, correspondant à l'ensemble complet de toutes les configurations possibles à la fin du jeu Tic-tac-toe où un joueur "*x*" jouant contre une joueur "*o*" a commencé le jeu. Les données sont décrites par 9 variables qualitatives.

Notre méthodologie expérimentale est la suivante : pour chaque jeu de données, nous avons considéré initialement un échantillon de taille 50 (pour la base Zoo), 100 (pour la base "*Auto*") et 700 (pour la base "*Tic-tac-toe*"). Sur cet échantillon, nous avons appliqué le processus de classification par b-coloration initial. La partition optimale générée est l'objet d'une mise à jour incrémentale en ajoutant successivement le reste des instances. La valeur de l'indice de *Dunn généralisé* est calculée après chaque ajout afin d'évaluer les performances de l'algorithme incrémental.

Pour une meilleure évaluation des résultats obtenues sur ces jeux de données, notre algorithme a été comparé avec l'approche de *b-coloration originale* (*i.e.* qui consiste à relancer pour chaque ajout tout le processus de classification par b-coloration décrit dans la section 3.3) et avec les approches les plus classiques de classification incrémentale, à savoir un algorithme de type *passage simple* ("Single Pass") et *l'algorithme des k-plus poches voisins* [Cover and Hart, 1967].

L'algorithme de *passage simple* traite les données séquentiellement en comparant chaque individu à toutes les classes existantes. Si la dissimilarité entre l'individu introduit au système et certaines classes est supérieure à un seuil fixé, alors l'individu est ajouté à la classe la plus proche; autrement il forme sa propre classe. L'algorithme des *k-plus poches voisins* mesure, quant à lui, pour chaque nouvel individu sa dissimilarité à tous les autres individus de la population, et identifie les $k$ individus les plus proches. La classe avec le label le plus fréquent dans l'échantillon des $k$ points sera ensuite attribué au nouvel individu.

Les courbes des figures 3.19, 3.20 et 3.21, indiquent l'évolution de la valeur de l'indice de Dunn généralisé des partitions fournies (par les différentes approches) en fonction du nombre d'individus pour respectivement les jeux de données "*Zoo*", "*Auto*" et "*Tic-tac-toe*".

Il s'avère clairement que l'algorithme incrémental réalise de meilleurs résultats que les *k-plus proches voisins* ($k$=5) et des résultats plus significatifs que ceux de l'algorithme de *passage simple* (avec un *seuil de dissimilarité* égale au seuil optimal $\theta_o$ associé à la partition initiale).

Il apparaît bien que l'approche incrémentale possède un avantage léger par rapport à l'algorithme de *b-coloration* initial (outre l'apport en temps d'exécution). En effet, les courbes relatives à cette approche sont, dans la majorité des cas meilleures que celles de l'approche initiale. Ceci souligne l'efficacité de la stratégie de transformation (recoloration), effectuée lors de l'ajout d'une nouvelle instance, qui permet un raffinement de la partition initiale en minimisant la *dissimilarité intraclasse* et en améliorant ainsi l'indice de *Dunn généralisé.*

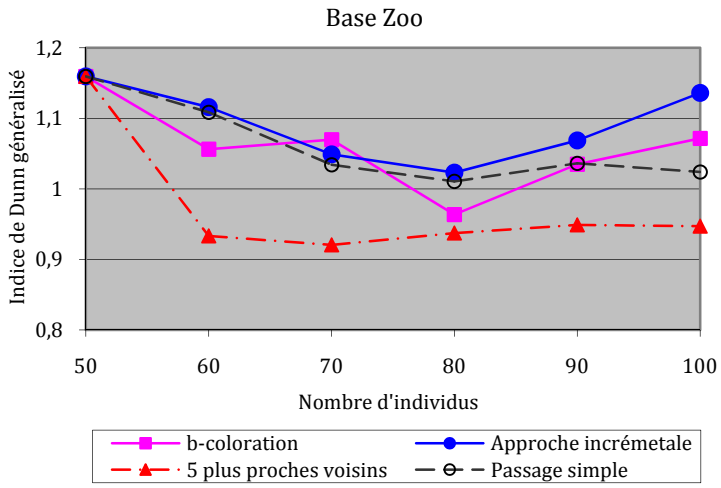



**Figure 3.19 - Performances sur la base Zoo**

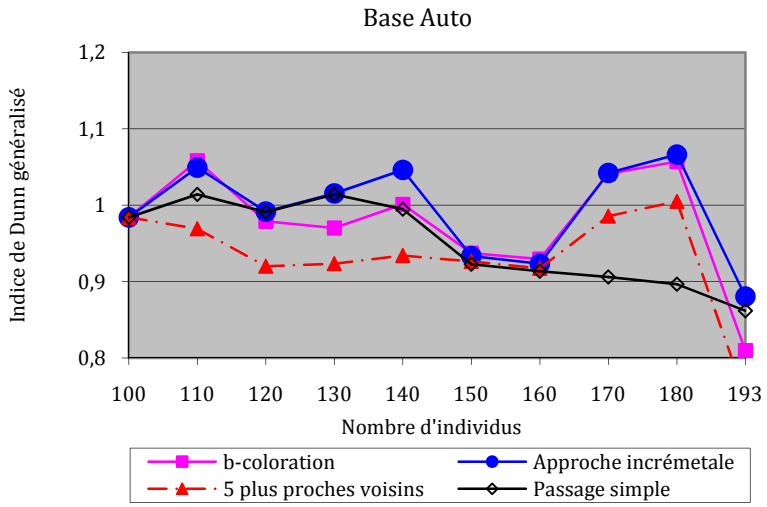



**Figure 3.20 - Performances sur la base Auto**

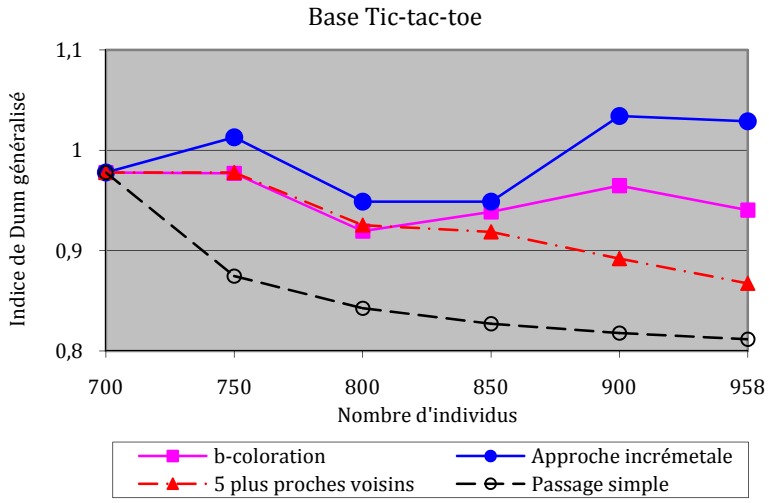



**Figure 3.21 - Performances sur la base Tic-tac-toe**

## 3.6. Conclusion

Ce chapitre nous a permis de présenter dans le détail la nouvelle approche de classification automatique sur tableau de dissimilarités que nous avons élaborée. Elle est basée sur une nouvelle technique de coloration de graphes, baptisée la *b-coloration*. Cette technique de coloration possède l'avantage de fournir une partition fine des données où la *séparation interclasse* est réalisée simultanément avec la *cohésion intraclasse*, quand le nombre de classes n'est pas fixé a priori. Elle possède également un ensemble de caractéristiques intéressantes, à savoir : (1) elle est applicable à tout type de données dès lors qu'il est possible de construire une matrice de dissimilarités entre les individus à classer, et (2) elle permet d'offrir une description des groupes trouvés par des points dominants qui d'une part sont le reflet des propriétés de leur classe (ce qui facilite l'interprétabilité des classes) et qui garantissent d'autre part de trouver des classes *significatives* et *bien-séparées*.

Nous avons présenté également l'application de cette nouvelle méthode de classification sur des jeux de données benchmark, sur une base d'images archéologiques et sur une base de données réelles du *Programme de Médicalisation des Systèmes d'Information* (PMSI). Pour cette dernière l'objectif était d'obtenir une partition fine constituée de groupes homogènes et bien séparés de séjours hospitaliers. Cette étude a permis d'identifier les groupes homogènes de malades particulièrement hétérogènes, et a ouvert de nouvelles perspectives dans ce domaine notamment pour construire de façon automatique des groupes homogènes et bien séparés à partir des données de l'année précédente. Cette démarche présente un réel intérêt pour les responsables de l'information médicale qui pourraient ainsi plus facilement affecter des forfaits ad-hoc pour le remboursement des GHM aux établissements, tentant ainsi de remédier à l'insatisfaction qui subsiste des deux côtés (établissements et administration).

Nous avons ensuite proposé une extension à notre approche de *classification non supervisée* qui concerne l'apprentissage automatique. Un nouvel algorithme de mise à jour incrémentale, se basant uniquement sur la connaissance des dissimilarités entre les individus pris deux à deux et sur la notion de dominance des classes, a ainsi été conçu. Il permet l'orientation d'un nouvel individu plongé dans le système vers la classe adéquate ou le réarrangement de la partition si des données existantes sont supprimées.

L'avantage de cet algorithme est qu'il effectue une classification dynamique qui satisfait les propriétés de la b-coloration et les performances de partitionnement en termes de qualité (l'indice de *Dunn généralisé*) et de temps d'exécution (complexité), et toujours sans que le nombre de classes n'ait été fixé à l'avance et sans limitation sur les type de données traitées.

# PARTIE 2

## ANALYSE DES DONNEES SEQUENTIELLES

## APPLICATION AUX TRAJECTOIRES HOSPITALIERES

# ÉTAT DE L'ART : ANALYSE DES DONNÉES SÉQUENTIELLES


## Résumé

*Ce chapitre présentera le cadre général de la deuxième partie de cette thèse : l'analyse de données séquentielles. Après l'identification des différentes tâches de l'analyse de données séquentielles, nous aborderons de près le problème de la classification automatique des séquences temporelles en évoquant les approches les plus répondues dans la littérature (les approches par proximité et les approches par modèles de mélange). Nous relatons pour chacune de ces approches, le principe, les avantages et les inconvénients. Cette synthèse permettra de présenter des approches alternatives de classification et de positionner notre contribution dans le chapitre suivant.*


## Sommaire

# Chapitre 4

# État de l'art : Analyse des données séquentielles

*" L'homme n'est pas une entité indépendante, mais un processus de construction directement inséré dans le flux temporel de son époque! "*

Norbert Elias, cité dans "Alan Turing" de Laurent Lemire

## 4.1. Introduction

Si la fouille de données, telle qu'elle est définie par Kodratoff *et al.* [Kodratoff *et al.*, 2001], peut s'apparenter à « *un processus interactif et itératif d'analyse d'un grand ensemble de données brutes afin d'en extraire des connaissances exploitables par l'analyste qui y joue un rôle central* », la fouille de données *séquentielles* elle, fournit en plus une capacité à suggérer les causes et les effets.

Nous pouvons identifier deux types d'objets sur lesquels les algorithmes de fouille de données séquentielles opèrent : (1) les *séries séquentielles* représentant des données provenant de sources continues, et (2) les *séquences temporelles* représentant des listes ordonnées d'évènements. C'est à ce dernier type de données que nous nous intéressons dans la suite de cette thèse.

Les principales sources statistiques des *séquences temporelles*, décrivant l'évolution d'individus à différentes échelles de temps, sont en général des enquêtes rétrospectives. Dans ces enquêtes sont recueillies, pour un échantillon d'individus, les trajectoires définies par les changements d'état de variables représentant leur comportement.

Au cours de la dernière décennie, différentes techniques de fouille de données séquentielles ont été proposées et se sont avérées utiles dans différents domaines d'application (marketing, sociologie, médecine). Comme la fouille de données séquentielles exploite des mécanismes et techniques propres à différentes disciplines, à savoir les *statistiques*, *l'apprentissage automatique* et les *bases de données*, les travaux proviennent de sources différentes [Antunes and Oliveira, 2001; Laxman and Sastry, 2006]. Cependant les principaux *objectifs* (*tâches* ou *opérations*) de la fouille de données

séquentielles sont les mêmes et peuvent être regroupés comme suit : (1) la *prévision*, (2) le *classement*, (3) la *découverte de motifs* et (4) la *classification automatique*. Nous proposons d'expliciter chacune de ces tâches dans les sections suivantes pour ensuite pouvoir situer notre approche.

**Prévision.** Encore appelée *prédiction* par abus de langage dans le cadre des séries temporelles (ou traduction littérale du terme anglais), la prévision consiste à évaluer (ou prévoir) l'état d'une séquence temporelle à un instant $t$ en ayant connaissance de ses valeurs aux instants $t_i<t$. Les approches de prévision dans les séquences temporelles opèrent généralement en deux étapes : dans un premier temps, elles cherchent à élaborer un modèle des données qui résume en quelque sorte les relations qui existent entre les différents états des séquences traitées, et qui permet dans une deuxième étape de réaliser des prévisions optimales. La prévision des séries temporelles est un problème traité de longue date par les méthodes statistiques classiques. Les modèles autorégressifs (ARMA, ARIMA) dus à Box et Jenkins en 1970 [Box and Jenkins, 1970] ont été utilisés avec succès dans plusieurs domaines d'applications (économiques, industrielles, etc.) à des fins de prévision des valeurs futures dans une séquence temporelle, par combinaison linéaire des valeurs précédentes [Box *et al.*, 1994 ; Chatfield, 1996; Hastie *et al.*, 2001]. Alors que les modèles ARMA (*AutoRegressive Moving Average*) ne permettent de traiter que les séquences temporelles stationnaires[8], les modèles ARIMA (*AutoRegressive Integrated Moving Average*) ont été proposés pour traiter les séquences non stationnaires après avoir déterminé le nombre de fois où il faut différencier la série avant de la rendre stationnaire. La limite des modèles autorégressifs réside essentiellement dans la nature des données : les valeurs qualitatives nominales rendent par exemple ces techniques inapplicables. Dans ce cadre, les modèles markoviens (*chaînes de Markov* ou *modèles de Markov cachés*) peuvent être utilisés pour réaliser des prévisions sur l'état futur du processus modélisé. Récemment des techniques statistiques non paramétriques telles que les réseaux de neurones ont également été proposées [Koskela *et al.*, 1996].

**Classement (ou aussi classification supervisée).** Considéré comme l'une des techniques de la fouille de données les plus anciennes et les plus souvent utilisées, que ce soit en *médecine* pour connaître la pathologie d'un patient, en *marketing* pour identifier le profil d'un client, en *sociologie* pour connaître la catégorie démographique d'un individu, le classement de données séquentielles suppose que chaque séquence temporelle d'une population donnée appartient à une classe (ou catégorie) prédéfinie et cherche à déterminer automatiquement la catégorie correspondante à toute nouvelle séquence introduite dans le système. Au cours des dernières années, plusieurs techniques de classement de données séquentielles ont été développées. Les plus répandues sont les approches à base de *prototypes* et les approches à base de *modèles*. Avec la première

[8] Une *séquence* ou *série temporelle* est dite *stationnaire* si ses réalisations sont issues d'un même processus aléatoire dont les paramètres (moyenne, variance, auto-corrélation…) restent les mêmes au cours du temps.

famille, chaque classe prédéfinie est caractérisée par une *séquence "type"* représentative de la classe (*prototype*), et le classement d'une nouvelle séquence s'opère en regardant parmi les prototypes des classes celui qui est le plus proche de cette séquence, au sens de la métrique choisie. Nous détaillons certaines mesures de ressemblance sur les séquences temporelles dans la section 4.2.1. Les techniques de classement de données séquentielles à base de *modèle*s quant à elles se basent souvent sur les modèles markoviens (chaines de Markov et *modèles de Markov cachés : HMM*). Ces modèles résument l'information contenue dans chaque classe et peuvent ensuite être appliqués à de nouvelles séquences pour en déduire leur affectation. Dans le cas des HMM par exemple, un algorithme d'apprentissage permet d'ajuster les paramètres du modèle de chaque classe. Une nouvelle séquence est ensuite assignée à la classe dont le modèle HMM est le plus susceptible de la reproduire.

**Découverte de motifs séquentiels.** Une autre technique descriptive, moins répandue que la classification automatique mais qui intéresse de plus en plus de secteurs stratégiques tels que le marketing, la finance, ou encore la médecine (identification des symptômes précédant les maladies) est l'*extraction automatique de motifs séquentiels*. L'opération d'extraction de tels motifs, introduit par Agrawal en 1995 [Agrawal and Srikant, 1995], peut être considérée comme une extension de l'extraction de règles d'association dans les bases de données transactionnelles. La recherche de motifs séquentiels consiste à extraire des séquences, c'est-à-dire des ensembles de symboles fréquemment associés sur une période de temps déterminée, et susceptibles de se reproduire. Ce mécanisme peut permettre d'identifier certains comportements typiques des individus dans le temps.

**Classification automatique.** Comme explicité dans les premiers chapitres de ce mémoire, la classification automatique cherche à identifier une typologie des individus en groupes homogènes et bien séparés. Appliquée aux séquences de données temporelles, les individus manipulés sont cette fois des séquences dont on cherche à mettre en relation statistique leurs différents enregistrements. De façon générale, la classification automatique est utile comme préalable à d'autres opérations de fouille de données. En effet, une démarche fréquente du *data mining* est de trouver des classes d'individus qui ont le même type de comportement, afin d'ensuite pouvoir identifier la classe à laquelle appartient un nouvel individu en exploitant son historique (*classement*) et évaluer ainsi son comportement futur (*prévision*). C'est pour ces différentes raisons que nous nous concentrons dans la suite de cette thèse sur les algorithmes exploratoires de données séquentielles pour la classification automatique.

## 4.2. Les approches de classification des données séquentielles

Plusieurs techniques de classification automatique des données séquentielles ont été développées ces dernières années. Elles ont été appliquées dans différents domaines comme l'analyse de séquences biologiques [Cadez *et al.*, 2000a], l'étude de la mobilité des

objets dans les vidéos [Buzan *et al.*, 2004], ou d'autres domaines pour lesquels l'individu joue un rôle central, à savoir la classification des patients sur la base de leurs groupes sanguins [Cadez *et al.*, 2000a], la modélisation du comportement d'utilisateurs sur le Web [Cadez *et al.*, 2000b], etc. Parmi cette variété de méthodes de classification, les plus répandues sont les approches *par proximité* et les approches *par modèles de mélange*.

## 4.2.1. Les approches de classification fondées sur la notion de proximité

### *4.2.1.1. Présentation générale*

De nombreuses méthodes en analyse des données s'appuient sur le concept de similarité et de distance entre les objets à analyser. Les approches de classification automatique des données séquentielles ont naturellement besoin de connaître la proximité entre les séquences pour les regrouper. Cette section présente différentes techniques d'évaluation de la proximité entre des séquences temporelles, qui pose notamment le problème de séquences de longueurs variables. Les distances les plus utilisées sont la *distance euclidienne*, *l'alignement temporel dynamique* (DTW : Dynamic Time Warping) et la *plus longue sous-séquence commune* (LCS : Longest Common Subsequence).

Une fois l'indice de proximité défini pour l'ensemble des séquences de la population, certaines approches classiques de classification automatique présentées dans le deuxième chapitre peuvent facilement être appliquées. C'est les cas notamment des approches hiérarchiques de classification, des approches des *k-médianes* et de celles fondées sur la théorie des graphes (voir chapitre 2). Ces approches requièrent l'utilisation d'une mesure de ressemblance entre les séquences (ou les trajectoires) sur la base de laquelle elles essaient de construire une partition de ces séquences en classes homogènes et bien séparées. Ces différents aspects sont détaillés dans les sections suivantes.

### *4.2.1.2. Quelques distances adaptées aux séquences temporelles*

#### *a. La distance euclidienne*

La distance euclidienne est une des distances les plus utilisées. Elle présente l'avantage d'être intuitive et simple à mettre en œuvre, cependant elle se trouve vite limitée face à des données bruitées, translatées, périodiques ou de longueurs différentes. La distance euclidienne $d(S_i, S_j)$ entre deux séquences temporelles $S_i = e_{i,1}, e_{i,2}, \dots, e_{i,T_i}$ et $S_j = e_{j,1}, e_{j,2}, \dots, e_{j,T_j}$ ($e_{i,t}$ est la $t^{ème}$ observation de la séquence $S_i$) de longueurs différentes ($T_i \neq T_j$) est défini comme suit :

$$d(S_i, S_j) = \sqrt{\sum_{t=1}^{min(T_i,T_j)} \left(e_{i,t} - e_{j,t}\right)^2} \quad (4.1)$$

### *b. Dynamic Time Warping (DTW)*

Pour pallier aux problèmes liés à la distance euclidienne, Sakoe [Sakoe, 1979] a introduit la distance *DTW : Dynamic Time Warping* (*litt.*, alignement temporel dynamique) dans le domaine de la reconnaissance de la parole. Celle-ci a été utilisée pour mesurer la ressemblance entre un mot quelconque prononcé par un locuteur humain et plusieurs mots de référence, permettant notamment de s'affranchir du rythme de prononciation.

La DTW [Kruskall and Liberman, 1999] est reconnue par la suite comme une mesure très fiable permettant d'évaluer la distance entre deux séquences de longueurs non nécessairement identiques, tout en prenant en compte l'effet de translation (dilatation) présent dans les données, c'est-à-dire la présence ou non d'états intermédiaires entre les états étudiés des deux séquences. Sémantiquement, pour comparer deux séquences temporelles avec la distance DTW, la démarche consiste à déformer les deux séquences en insérant des "-" (ceci revient concrètement à étirer l'une et/ou l'autre des séquences) jusqu'à l'obtention de la "meilleure" mise en correspondance (recouvrement) entre les séquences modifiées. Un tel processus est baptisé *l'alignement temporel*.

L'algorithme de calcul de la DTW réalise ainsi cet alignement en recherchant, parmi tous les alignements possibles, celui qui minimise une fonction de coût $\gamma$ intégrant l'écart entre les données alignées et un coût de déformation temporelle. La distance retenue est celle correspondant à l'alignement de coût minimal.

**<u>Implémentation algorithmique</u>**

Soit deux séquences temporelles $S_i = e_{i,1}, e_{i,2}, \dots, e_{i,T_i}$ et $S_j = e_{j,1}, e_{j,2}, \dots, e_{j,T_j}$ à comparer. La distance DTW entre $S_i$ et $S_j$ peut être déterminée par l'algorithme de programmation dynamique, de complexité $O(T_i, T_j)$, où $C[\,][\,]$ représente la matrice de cumul des distances et $d(e_{i,u}, e_{j,v})$ l'écart entre $e_{i,u}$ et $e_{j,v}$ donné par $|e_{i,u} - e_{j,v}|$.

***Entrée***: *Deux séquences* $S_i = e_{i,1}, e_{i,2}, \dots, e_{i,T_i}$ *et* $S_j = e_{j,1}, e_{j,2}, \dots, e_{j,T_j}$
***Sortie***: *La distance DTW entre* $S_i$*et* $S_j$: $DTW(S_i, S_j)$

*Créer une matrice* $C[0 \dots T_i][0 \dots T_j]$
$C[0][0] \leftarrow 0$;
$C[0 \dots T_i][0] \leftarrow \infty$;
$C[0][0 \dots T_j] \leftarrow \infty$;
***pour*** $u = 1$ à $T_i$ ***faire***
    ***pour*** $v = 1$ à $T_j$ ***faire***
        $coût \leftarrow d(e_{i,u}, e_{j,v})$;
        $C[u][v] \leftarrow coût + min\{C[u-1][v-1], C[u][v-1], C[u-1][v]\}$
    ***fin pour***
***fin pour***
***Retourner*** $(C[T_i][T_j])$

**L'algorithme DTW** (Dynamic Time Warping)

## Exemple

Pour illustrer le fonctionnement de l'algorithme, prenons l'exemple des deux séquences $S_1 = 3,6,8,7,8$ et $S_2 = 3,5,6,5,6,8,7$. Le tableau 4.1 illustre la matrice de cumul des distances $C[\,][\,]$ de taille 7x5. La dernière case $C[7][5]$ située en bas à droite de cette matrice et de valeur égale à 3, correspond à la distance $DTW(S_i, S_j)$. La mise en correspondance entre ces deux séquences est obtenue à partir de la matrice $C[\,][\,]$ en suivant le chemin qui a fourni la valeur 3. Un tel chemin est dit *chemin d'alignement optimal* et est représenté dans le tableau 4.1 par les cases grisées en gras.

Un alignement de coût minimal peut être donc le suivant :

| $S_1$: | 3 | - | - | - | 6 | 8 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| | \| | \| | \| | \| | \| | \| | \| | \| |
| $S_2$ : | 3 | 5 | 6 | 5 | 6 | 8 | 7 | - |

| | | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|
| | $S_1$ / $S_2$ | | **3** | **6** | **8** | **7** | **8** |
| 0 | | 0 | ∞ | ∞ | ∞ | ∞ | ∞ |
| 1 | **3** | ∞ | **0** | 3 | 8 | 12 | 17 |
| 2 | **5** | ∞ | 2 | **1** | 4 | 6 | 9 |
| 3 | **6** | ∞ | 5 | **1** | 3 | 4 | 6 |
| 4 | **5** | ∞ | 7 | **2** | 4 | 5 | 7 |
| 5 | **6** | ∞ | 10 | **2** | 4 | 5 | 7 |
| 6 | **8** | ∞ | 15 | 4 | **2** | 3 | 3 |
| 7 | **7** | ∞ | 19 | 5 | 3 | **2** | **3** |

**Tableau 4.1 - Matrice de cumul de distances pour le calcul de DTW**

### c. La plus longue sous-séquence commune (LCS)

Proposée initialement pour la comparaison de chaînes de caractères, la mesure de la *plus longue sous-séquence commune* (*LCS*, *Longest Common Subsequence*) de Paterson [Paterson and Dancik, 1994] a été considérée par la suite comme un cas particulier de la *Dynamic Time Warping* spécifique aux données qualitatives (catégorielles). Utilisant le même principe que la DTW, l'algorithme de recherche de la plus longue sous-séquence commune réduit la distance de cumul pour chaque comparaison entre les symboles des séquences à 1 ou 0, selon la présence ou l'absence du même symbole.

**Définition**

Soient $S_1$ et $S_2$ deux séquences de données catégorielles (dites chaînes de caractères). Une sous-séquence commune à $S_1$ et $S_2$ est une chaîne de caractères $c$ dont les éléments apparaissent à la fois dans $S_1$ et $S_2$ en respectant l'ordre préétabli dans ces deux séquences. Nous notons $LCS(S_1, S_2)$, la longueur maximale d'une sous-séquence commune à $S_1$ et $S_2$.

Le problème de l'évaluation de la *distance* entre deux chaînes de caractères est une généralisation du problème de l'évaluation de la longueur d'une plus longue sous-séquence commune à ces deux chaînes de caractères. Cette distance appelée *distance d'édition* est un moyen typique des approches de la reconnaissance d'écriture manuscrite, mais elle a été aussi utilisée pour mesurer la quantité d'évolutions séparant deux séquences biologiques et dans la classification automatique de différents types de trajectoires [Buzan *et al.*, 2004].

Plus précisément, la distance d'édition entre deux séquences de données catégorielles $S_i$ et $S_j$ s'écrit alors :

$$d_E(S_i, S_j) = |S_i| + |S_j| - 2 * LCS(S_i, S_j) \quad (4.2)$$

**Implémentation algorithmique**

La mesure de la plus longue sous-séquence commune à deux séquences de données catégorielles peut être également calculée par un algorithme de programmation dynamique, de complexité $O(T_i, T_j)$, de la façon suivante (la matrice de cumul est ici appelée $L$) :

***Entrée***: *Deux séquences de données catégorielles* $S_i = e_{i,1}, e_{i,2}, \dots, e_{i,T_i}$ *et* $S_j = e_{j,1}, e_{j,2}, \dots, e_{j,T_j}$
***Sortie***: *La longueur maximale d'une sous − séquence commune à* $S_i$ *et* $S_j$ : $LCS(S_i, S_j)$

```
1:  Créer une matrice L[0 ... T_i][0 ... T_j]
2:  L[0][0] ← 0;
3:  L[0 ... T_i][0] ← 0;
4:  L[0][0 ... T_j] ← 0;
5:  pour u = 1 à T_i faire
6:      pour v = 1 à T_j faire
7:          si (e_{i,u} = e_{j,v}) alors
8:              L[u][v] ← L[u − 1][v − 1] + 1;
9:          fin si
10:         sinon
11:             si (L[u − 1][v] > L[u][v − 1]) alors
12:                 L[u][v] ← L[u − 1][v];
13:             sinon
14:                 L[u][v] ← L[u][v − 1];
15:             fin si
16:         fin si
17:     fin pour
18: fin pour
19: Retourner (L[T_i][T_j])
```

**Algorithme** de recherche de la plus longue sous-séquence commune

## Exemple

Soit les deux séquences $S_1 = CATCAGTA$ et $S_2 = ACTCCATGCA$. Le tableau 4.2 illustre la matrice de cumul des distances $L[\,][\,]$ de taille 9x11. Nous avons $LCS(S_1, S_2) = 6$ et $d_E(S_1, S_2) = |S_1| + |S_2| - 2 * LCS(S_1, S_2) = 7 + 10 - 2 * 6 = 6$.

Des sous-séquences maximales communes à $S_1$ et $S_2$ sont par exemple $CTCAGA$ et $ATCATA$.

| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $S_2$ \ $S_1$ | | **A** | **C** | **T** | **C** | **C** | **A** | **T** | **G** | **C** | **A** |
| 0 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | **C** | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | **A** | 0 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| 3 | **T** | 0 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 4 | **C** | 0 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 4 | 4 |
| 5 | **A** | 0 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 4 | 4 | 5 |
| 6 | **G** | 0 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 5 | 5 | 5 |
| 7 | **T** | 0 | 1 | 2 | 3 | 3 | 3 | 4 | 5 | 5 | 5 | 5 |
| 8 | **A** | 0 | 1 | 2 | 3 | 3 | 3 | 4 | 5 | 5 | 5 | **6** |

**Tableau 4.2 - Matrice de cumul de distances L pour le calcul de LCS**

#### *4.2.1.3. Conclusion*

Les approches de classification automatique fondées sur un indice de proximité sont particulièrement adaptées à la recherche des différents *profils* d'individus constituant la population. Pour cela, elles cherchent à découvrir une partition des données en classes homogènes et bien séparées, de sorte que les séquences les plus proches (au sens de la métrique utilisée) se retrouvent dans une même classe (*cohésion intraclasse*), alors que les séquences dissemblables sont rattachées à des classes différentes (*séparation interclasse*). Néanmoins, les classes obtenues par les approches de classification fondées sur un indice de proximité ne sont pas toujours facilement interprétables. En effet, la plupart de ces méthodes arrivent souvent à fournir une description des classes via des séquences dites "types" (à savoir, les séquences centrales des classes par exemple) mais échouent pour élaborer des modèles résumant les informations contenues dans les séquences de la classe et les relations qui existent entre elles. Or pour beaucoup d'applications d'aide à la décision, il semble nécessaire d'être capable de décrire les classes de la population sous une forme compacte permettant une éventuelle abstraction des données. En conséquence, il est difficile avec ces méthodes de prendre en compte les nouvelles séquences introduites dans le système pour en déduire leurs classes et prévoir la suite du comportement de leurs individus correspondants.

### 4.2.2. Les approches de classification par modèles de mélange

#### *4.2.2.1. Mélange de densité*

En statistiques, nous appelons mélange de densité une fonction de densité de probabilité qui est modélisée par une combinaison linéaire de plusieurs fonctions de densité composantes.

L'estimation d'une fonction de densité d'une variable $s$ est en réalité donnée par la modélisation d'une densité de probabilité $P(s)$ à partir d'un échantillon de données observées, supposées issues de cette densité. Pour un mélange, la densité de probabilité est fonction des densités composantes, et elle est définie par :

$$P(s) = \sum_{h=1}^{k} P(s|c) * P(c) \tag{4.3}$$

où :

- $k$ est le nombre de composantes du mélange.
- $\{P(c), c = 1 \dots k\}$ est une famille de scalaires représentant les paramètres du mélange. $P(c)$ est la probabilité a priori pour qu'une observation ait été générée par la composante $h$ du mélange, sachant que $\sum_{c=1}^{k} P(c) = 1$,
- $P(s|c)$ sont les densités composantes.

#### *4.2.2.2. Classification par mélange de densités (CMD)*

L'utilisation du modèle de mélange de probabilité en classification automatique est devenue aujourd'hui une approche classique. Etant donné un ensemble de données $S = \{S_1, S_2, \dots, S_n\}$ (avec $S_i = \{s_{i,1}, s_{i,2}, \dots, s_{i,n_i}\}$ l'ensemble des $n_i$ trajectoires $s_{i,j}$ observées pour l'individu $i$) et un nombre de classes $k$ fixé a priori, Cadez *et al.* [Cadez *et al.*, 2000a] ont proposé un cadre générique de classification automatique des données séquentielles. Celui-ci est basé sur un modèle de mélange de densités et son principe général consiste à :

- sélectionner un individu *i* de la population,
- l'individu *i* est attribué à l'une des $k$ classes ($c = 1, \dots, k$) de probabilité $P(c)$ tel que $\sum_{c=1}^{k} P(c) = 1$,
- à chaque classe $c$ correspond un modèle de génération de données $P(S_i|c_i = c, \phi_c)$, où $\phi_c$ sont les paramètres de cette distribution de probabilité, $S_i$ est la donnée de l'individu $i$ et $c_i$ désigne la classe de l'individu *i*. Ce modèle permet en pratique de calculer la probabilité qu'un individu ait un vecteur $S_i$ de données, sachant qu'il appartient à la classe $c$.

Selon ces différentes hypothèses, chaque individu $i$ est attribué à une classe $c$ ($1 \leq c \leq k$). En supposant que les observations de l'individu $i$ sont conditionnellement indépendantes, et connaissant les paramètres du modèle de la classe $c$, $S_i$ possède la densité de probabilité suivante :

$$P(S_i|c_i = c, \phi_c) = P(S_i|\phi_c) = \prod_{j=1}^{n_i} P(s_{i,j}|\phi_c) \tag{4.4}$$

Pour le problème de classification automatique, les auteurs supposent que chaque observation $S_i$ est issue d'un mélange de densité et ils cherchent à trouver les paramètres du modèle qui maximisent la vraisemblance des $n$ observations $S = \{S_1, S_2, \dots, S_n\}$, en supposant que ces observations sont indépendantes.

D'après l'équation 4.3, la distribution de probabilité de $S_i$ dont la classe $c_i$ étant inconnue est une fonction linéaire des modèles composants. Elle est de la forme :

$$P(S_i|\phi) = \sum_{c=1}^{k} P(S_i|c_i = c, \phi_c) * P(c) \tag{4.5}$$

où $\phi = \{\phi_1, \phi_2, \dots, \phi_k\}$ est l'ensemble des paramètres des classes $c$ ($1 \leq c \leq k$).

En considérant maintenant que les individus sont indépendants, la vraisemblance totale de l'ensemble des données $S = \{S_1, S_2, \dots, S_n\}$ est donnée par l'équation suivante :

$$P(S|\phi) = \prod_{i=1}^{n} P(S_i|\phi) = \prod_{i=1}^{n} \sum_{c=1}^{k} P(S_i|c_i = c, \phi_c) * P(c) \quad (4.6)$$

Comme illustration au cadre générique proposé, Cadez *et al.,* [Cadez *et al.*, 2000b] ont appliqué un mélange de *chaînes de Markov* pour tenter de modéliser le comportement des utilisateurs sur le Web, comme nous le verrons ci-après. Dans ce cas, l'estimation des paramètres des modèles $\phi = \{\phi_1, \phi_2, \dots, \phi_k\}$ maximisant la vraisemblance des $n$ observations $S = \{S_1, S_2, \dots, S_n\}$ se fait par un algorithme itératif de type *Espérance-Maximisation* EM (*Expectation-Maximisation* en anglais) [Dempster *et al.*, 1977]. Certains auteurs ont proposé également d'utiliser le mélange de *modèles de Markov caché* (*Hidden Markov Models, HMM*) pour la classification automatique des séquences. Ceux-ci adoptent souvent le formalisme des *nuées dynamiques* afin d'estimer les classes des individus et ajuster les paramètres de leurs modèles.

### *a. Classification par mélange de chaînes de Markov*

Baptisés les *modèles de Markov observables*, les *chaînes de Markov* sont des automates probabilistes à états finis. Ils se basent sur l'hypothèse de Markov : "*le futur ne dépend que du présent et non du passé*". Un modèle de Markov est alors un processus aléatoire qui peut changer d'état $e$ $(1 \leq e \leq m)$ au hasard aux instants $t = 1,2, \dots T$.

Une évolution du système est donc une suite de transitions d'états $e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow \cdots \rightarrow e_T$ à partir d'un état de départ $e_1$.

Le système émet cette séquence $e_1, e_2, \dots, e_T$ avec une probabilité $P(e_1, e_2, \dots, e_T)$ définie de proche en proche, comme suit :

$$\begin{aligned} P(e_1, e_2, \dots, e_T) &= P(e_1, e_2, \dots, e_{T-1}) * P(e_T|e_1, e_2, \dots, e_{T-1}) \\ &= P(e_1) * P(e_2|e_1) * P(e_3|e_2e_1) * \cdots * P(e_T|e_1, e_2, \dots, e_{T-1}) \end{aligned} \quad (4.7)$$

Ainsi, pour calculer $P(e_1, e_2, \dots, e_T)$, il suffit de se donner la probabilité initiale $P(e_1)$ et les probabilités des états conditionnés par les évolutions antérieures. Etant donné que ce processus aléatoire vérifie la propriété de Markov définie précédemment, nous avons donc :

$$\forall t, P(e_t|e_1, e_2, \dots, e_{t-1}) = P(e_t|e_{t-1}) \quad (4.8)$$

La probabilité $P(e_1, e_2, \dots, e_T)$ de la séquence $e_1, e_2, \dots, e_T$ est déterminée ainsi par la probabilité de l'état initial et celles des transitions successives comme suit :

$$P(e_1, e_2, \dots, e_T) = P(e_1) * P(e_2|e_1) * P(e_3|e_2) * \cdots * P(e_T|e_{T-1}) \quad (4.9)$$

Ceci conduit à caractériser un modèle de Markov par l'ensemble de deux paramètres $(\pi, A)$ suivants :

- $\pi$ est un vecteur de probabilité d'état initial $[\pi(e)]_{1\leq e\leq m}$ avec $\pi(e)$ est la probabilité d'émettre le symbole $e$ à l'instant $t = 1$,
- $A$ est une matrice $m * m$ des probabilités de transitions $[a(e,\acute{e})]_{1\leq e,\acute{e}\leq m}$ avec $a(e,\acute{e})$ est la probabilité d'observer le symbole $\acute{e}$ à l'instant $t+1$ $\forall t$ après avoir émis le symbole $e$ à l'instant $t$.

Le problème de classification automatique d'un ensemble de $n$ données observées $S = \{S_1, S_2, \dots, S_n\}$ tel qu'il est défini par Cadez *et al.* [Cadez *et al.*, 2000a] est ramené à un problème de classification statistique de *séquences temporelles*, utilisant des chaînes de Markov. L'utilisation des modèles markoviens permet de décrire la dynamique, *i.e.* la séquence d'états des trajectoires dans chaque classe. Par conséquent, on dira que des trajectoires sont similaires si elles sont produites par le même modèle et vice-versa.

Pour faciliter la compréhension des différentes étapes de l'algorithme de classification par mélange de chaînes de Markov [Cadez *et al.*, 2000a], nous allons supposer que chaque individu $i$ n'a qu'une seule trajectoire observée ($i.e.\ n_i = 1$). Soit $S_i = e_{i,1}, e_{i,2}, \dots, e_{i,T_i}$ une telle trajectoire avec $e_{i,t}$ est le symbole de la séquence $S_i$ observé à l'instant $t$ de l'espace des $m$ états possibles. Ainsi, de la définition d'une chaîne de Markov, l'équation 4.4 devient :

$$P(S_i|c_i = c, \phi_c) = P(S_i|\phi_c) = \pi_c(e_{i,1}) * \prod_{t=2}^{T_i} a_c(e_{i,t-1}, e_{i,t}) \qquad (4.10)$$

où $\phi_c = (\pi_c, A_c)$ sont les paramètres du modèle de Markov correspondant à la classe $c$.

Les équations (4.5), (4.6) et (4.10) spécifient complètement le modèle à partir des données observées $S = \{S_1, S_2, \dots, S_n\}$. Nous modélisons d'abord les trajectoires individuelles (équations 4.5 et 4.10) et ensuite les données pour l'ensemble des individus (équation 4.6).

Afin d'estimer les paramètres du modèle $\phi_c = (\pi_c, A_c)$ de chaque classe $c$ de la partition ($c = 1, \dots, k$), l'algorithme *Espérance-Maximisation* (EM) est appliqué. Au sens de l'algorithme EM [Dempster *et al.*, 1977], cette estimation est considérée comme un problème de données manquantes constituées par les classes des $n$ individus (on parle également de données latentes ou non observées). L'idée de base de cet algorithme consiste à raisonner sur les données complètes (données observées + données latentes) tout en prenant en compte le fait que l'information disponible sur les données latentes ne peut venir que des données observées (*i.e.* les séquences ou les trajectoires). Dans le cas de mélange de modèles markoviens, ceci se traduit par un algorithme itératif où chaque itération procède en deux étapes :

**Espérance**

Dans cette étape, l'algorithme calcule les probabilités conditionnelles d'appartenance aux classes de chaque individu $P(c_i = c|S_i, \phi)$, pour chacun des $k$ modèles de classe avec les paramètres courants $\phi$.

$$P(c_i = c|S_i, \phi) = \frac{P(S_i|c_i = c, \phi_c) * P(c)}{\sum_{u=1}^{k} P(S_i|c_i = u, \phi_u) * P(u)} \qquad pour\ 1 \leq c \leq k \tag{4.11}$$

**Maximisation**

Cette étape de maximisation vise à ajuster les différents paramètres des $k$ modèles de classe $\phi_c = (\pi_c, A_c)$, en pondérant chaque individu $i$ par sa probabilité conditionnelle d'appartenance aux classes, $P(c_i = c|S_i, \phi)$. En pratique, ces équations estiment que les nouvelles proportions du mélange (*i.e.* les probabilités a priori des composantes $P(c)$) sont proportionnelles aux probabilités d'appartenance aux classes $\mathrm{P}(c_i = c|S_i, \phi)$ (voir équation 4.12), alors que les nouvelles probabilités d'états initiaux $\pi_c$ (équation 4.13) et des probabilités de transition $A_c$ (équation 4.14) sont obtenues en comptant les états initiaux et les transitions, et en les pondérant par les probabilités d'appartenance au classes.

$$P^{Nouveau}(c) = \frac{1}{n}\sum_{i=1}^{n} P(c_i = c|S_i, \phi) \qquad pour\ 1 \leq c \leq k \tag{4.12}$$

$$\pi_c^{Nouveau}(e) = \frac{\sum_{i=1}^{n} P(c_i = c|S_i, \phi) * \delta(e, e_{i,1})}{\sum_{i=1}^{n} P(c_i = c|S_i, \phi)} \qquad pour \begin{matrix} 1 \leq c \leq k \\ 1 \leq e \leq m \end{matrix} \tag{4.13}$$

avec $\delta(e, e_{i,1}) = \begin{cases} 1, & si\ e = e_{i,1} \\ 0, & sinon \end{cases}$

$$a_c^{Nouveau}(e, é) = \frac{\sum_{i=1}^{n} P(c_i = c|S_i, \phi) * r_i^{e \to é}}{\sum_{i=1}^{n} P(c_i = c|S_i, \phi) * r_i^{e \to}} \qquad pour \begin{matrix} 1 \leq c \leq k \\ 1 \leq e, é \leq m \end{matrix} \tag{4.14}$$

avec :

- $r_i^{e \to é}$ est le nombre de transitions de $e$ à $é$ dans la séquence observée pour l'individu $i$.
- $r_i^{e \to}$ est le nombre de transitions de $e$ à n'importe quel état dans la séquence observée pour l'individu $i$.

L'algorithme EM est itératif. Ainsi, une fois les paramètres des classes estimés à l'issue de l'étape de maximisation, il s'agit d'évaluer les nouvelles densités de probabilités $P(S_i|c_i = c, \phi_c)$ des données $S_i$ décrites par l'équation 4.10. Celles-ci vont servir à évaluer

les nouvelles probabilités conditionnelles d'appartenance aux classes de chaque individu $P(c_i = c|S_i, \phi)$ à l'étape E de l'itération suivante (équation 4.11).

L'algorithme cherche à accroître la vraisemblance des $n$ observations, calculée par l'équation 4.6 et ce jusqu'à ce qu'un maximum soit atteint. Ceci revient à trouver la valeur des paramètres $\phi$ qui donne la plus grande vraisemblance aux données observées. Soulignons qu'il est souvent plus facile de chercher $\phi$ en maximisant le logarithme de la fonction de vraisemblance du mélange. Cela conduit au même résultat qu'avec le critère initial (équation 4.6), puisque la fonction logarithme est monotone croissante.

Le principal avantage du cadre probabiliste réside dans le fait qu'il permet de tenir compte de la variabilité des séquences des individus dans l'estimation des paramètres des modèles de classes. En effet, si un individu $i$ génère une séquence plus longue que celles des autres individus de la population, alors sa séquence $S_i$ contribuera plus dans l'évaluation du paramètre $r_i$ de l'équation 4.14 et ainsi dans l'estimation des probabilités de transitions.

Ce cadre générique de classification par mélange de chaînes de Markov a été appliqué avec succès dans plusieurs domaines. Dans [Cadez *et al.*, 2000b], les auteurs ont appliqué la méthode pour modéliser le comportement des internautes sur le Web. Dans ce cas d'application, les trajectoires décrivent les sessions de connexion sur un site Web modélisées sous la forme d'une liste de symboles représentant des catégories de pages Web. L'estimation des paramètres des modèles $\phi = \{\phi_1, \phi_2, \dots, \phi_k\}$ maximisant la vraisemblance des $n$ observations $S = \{S_1, S_2, \dots, S_n\}$ a été obtenue avec l'algorithme EM. Le but de la classification de ces trajectoires est de permettre au concepteur du site Web, via l'analyse des classes obtenues, de prendre les mesures nécessaires pour améliorer le site web. Dans le domaine sociologique, *Estacio et al.* ont utilisé dans [Estacio-Moreno *et al.*, 2004] l'approche de classification automatique à base du mélange de chaînes de Markov pour des données biographiques issues d'une enquête effectuée à Cali (Colombie) en 1998. La démarche a notamment conduit à des résultats pertinents pour l'analyse de la mobilité résidentielle.

### *b. Classification par mélange de modèles de Markov cachés*

Le modèle de Markov caché [Rabiner, 1989] est un modèle de Markov où les états ne sont pas des évènements directement observables, mais des états virtuels représentant une certaine combinaison d'événements réels et qui ont ainsi des probabilités d'émissions de tels évènements. C'est le cas d'un processus aléatoire double puisqu'il est constitué d'une variable *cachée* (ou interne) représentant l'état du système et d'une autre variable observable appelée *observation* qui est conséquence de cet état interne.

Les modèles de Markov caché combinent les propriétés à la fois des distributions de probabilités et d'une machine à états. Ces propriétés en font une des modélisations les plus efficaces actuellement en reconnaissance de la parole et qui permettent de modéliser

des processus stochastiques variant dans le temps et de bien capturer la variabilité temporelle des séquences.

Formellement, un modèle de Markov caché noté $\phi = (A, B, \pi)$ est définit par les paramètres suivants :

- Ses états cachés, en nombre $R$, qui composent l'ensemble $S_E = \{s_1, s_2, \dots, s_R\}$. L'état où se trouve le HMM à l'instant $t$ est noté $q_t (\, q_t \in E)$.
- $m$ symboles observables dans chaque état. L'ensemble des observations est noté $V = \{v_1, v_2, \dots, v_m\}$. Un élément $O_t$ d'une séquence $O$ prenant sa valeur dans $V$ désigne un symbole observé à l'instant $t$.
- Une matrice $A$ de probabilités de transition entre les états du modèle :

$$a_{i,j} = A(i,j) = P\big(q_{t+1} = s_j \big| q_t = s_i\big) \quad pour \begin{matrix} 1 \le i,j \le R \\ \forall t \end{matrix} \tag{4.15}$$

avec $a_{i,j} \ge 0 \;\; \forall i,j \;\; et \;\; \sum_{j=1}^{R} a_{i,j} = 1$

- Une matrice $B$ de probabilités d'observation : $b_j(s)$ est la probabilité d'observer le symbole $v_s$ quand le modèle se trouve dans l'état $j$, soit :

$$b_j(s) \; = P\big(O_t = v_s \big| q_t = s_j\big) \;\; pour \begin{matrix} 1 \le j \le R \\ 1 \le s \le m \end{matrix} \tag{4.16}$$

avec $b_j(s) \ge 0 \;\; \forall j,s \;\; et \;\; \sum_{s=1}^{m} b_j(s) = 1$

- Un vecteur π de densités de probabilité initiales : $\pi = \{\pi_i\}_{i=1,2,\dots,R}$. $\pi_i$ représente la probabilité que l'état de départ du modèle soit l'état $i$, soit :

$$\pi_i \; = P(q_1 = s_i) \quad pour \; 1 \le i \le R \tag{4.17}$$

avec $\pi_i \ge 0 \;\; \forall i \;\; et \;\; \sum_{i=1}^{R} \pi_i = 1$

Un HMM peut être vu donc comme un processus permettant de générer une séquence d'observations $O_1 \to O_2 \to O_3 \to \cdots \to O_T$ comme suit :

1. choix de l'état de départ : $t = 1$ , choisir l'état initial $q_1 = s_i$ avec $\pi_i$ ;
2. observation dans l'état sélectionné : choisir l'observation $O_t = v_s$ avec $b_i(s)$ ;
3. sélection de l'état suivant : passer à l'état suivant $q_{t+1} = s_j$ à l'aide de $a_{i,j}$;
4. changement d'état : $t = t + 1$ ; si $t < T$, alors retourner à l'étape 2 sinon STOP.

Pour faciliter la compréhension de ces différentes notions, considérons le HMM de la figure 4.1 suivante. Ce HMM montre un modèle de prévision météorologique qui possède $R = 3$ états (Beau, Mauvais ou Variable). Le vecteur de densités de probabilité initiales, $\pi = \{0.5, 0.2, 0.3\}$, indique que le HMM peut être initialement dans l'état "Beau" avec une probabilité de 0.5, dans l'état "Mauvais" avec une probabilité de 0.2 et dans l'état

"Variable" avec une probabilité de 0.3 $\left(\sum_{i=1}^{R} \pi_i = 1\right)$. A tout instant $t$, un des $m = 3$ symboles dans $V = \{\text{Ensoleillé, Burmeux, Pluvieux}\}$ peut être choisi puis émis en analysant la matrice $B$ de probabilités d'observation $B = \left\{b_j(s) \middle| 1 \leq j \leq R, 1 \leq s \leq m\right\}$. A titre d'illustration, la modèle possède une forte probabilité (0.7) d'émettre le symbole "Ensoleillé" quand il se trouve à l'état "Beau".

Une fois un symbole de sortie choisi et émis par le modèle, le HMM passe à un nouvel état, sélectionné à partir de la matrice de probabilités de transition entre les états du modèle $A = \left\{a_{i,j} \middle| 1 \leq i, j \leq R\right\}$. Par exemple, la probabilité de passage de l'état "Beau" à l'état "Mauvais" est $a_{Beau,Mauvais} = 0.2$, alors que la probabilité de rester dans l'état "Beau" est $a_{Beau,Beau} = 0.5$, sachant que la somme des probabilités de transitions pour chaque état vaut $1\left(\sum_{j=1}^{R} a_{i,j} = 1\right)$.

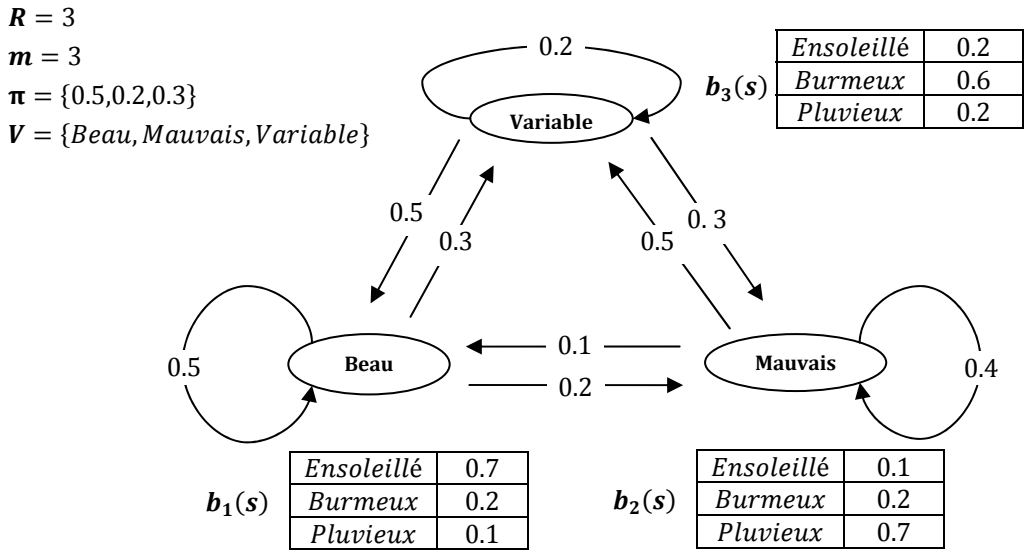



**Figure 4.1- Un exemple de modèle de Markov caché (HMM)**

Ce type de modèle peut être utilisé pour résoudre les trois problèmes suivants :

**Problème 1.** (*Estimation*) : évaluation de la probabilité de l'observation d'une séquence. Etant donnée la suite d'observations $O = \{O_1, O_2, \dots, O_T\}$ et un HMM $\phi = (A, B, \pi)$, comment évaluer la probabilité $P(O|\phi)$ d'apparition de cette séquence $O$ connaissant le modèle $\phi$ ? *Utilisation de l'algorithme Forward-Backward* [Baum *et al.*, 1970].

**Problème 2.** (*Explication*) : recherche du chemin le plus probable, ou estimation de la partie cachée. Soit la suite d'observations $O = \{O_1, O_2, \dots, O_T\}$ et un modèle $\phi = (A, B, \pi)$, comment trouver la suite d'états $Q = \{q_1, q_2, \dots, q_T\}$ qui explique le mieux $O$ ? *Utilisation de l'algorithme de Viterbi* [Forney, 1973 ]. Par exemple, la séquence produite par le modèle de la figure 4.1 est une séquence d'évènements.

La question est de savoir quelle séquence d'états a produit cette séquence d'évènements.

**Problème 3.** (*Apprentissage*) : comment ajuster les paramètres du modèle $\phi = (A, B, \pi)$, pour maximiser $P(S|\phi)$, à partir d'une ensemble de séquences d'apprentissage $S = \{S_1, S_2, \dots, S_n\}$ ( $S_l = e_{l,1}, e_{l,2}, \dots, e_{l,T_l}$) qui ont été émises par ce modèle ? *Utilisation de l'algorithme de Baum Welch* : un algorithme de type Espérance-Maximisation [Bilme, 1998] qui considère à chaque itération les formules de ré-estimation suivantes :

$$\pi_i^{Nouveau} = \frac{\sum_{l=1}^{n} \gamma_1^l(i)}{n} \qquad pour\ 1 \le i \le R \tag{4.18}$$

$$a_{i,j}^{Nouveau} = \frac{\sum_{l=1}^{n} \left( \sum_{t=1}^{T_l - 1} \left( \xi_t^l(i,j) \right) \right)}{\sum_{l=1}^{n} \left( \sum_{t=1}^{T_l - 1} \gamma_t^l(i) \right)} \qquad pour \begin{matrix} 1 \le i \le R \\ 1 \le j \le R \end{matrix} \tag{4.19}$$

$$b_j^{Nouveau}(s) = \frac{\sum_{l=1}^{n} \left( \sum_{t=1 \dots T_l - 1 \text{ où } v_s = e_{l,t}} \gamma_t^l(j) \right)}{\sum_{l=1}^{n} \left( \sum_{t=1}^{T_l - 1} \gamma_t^l(j) \right)} \qquad pour \begin{matrix} 1 \le j \le R \\ 1 \le s \le m \end{matrix} \tag{4.20}$$

où :

- $\xi_t^l(i,j)$ est la probabilité étant donnés une séquence d'observation $S_l$ et un HMM $\phi$, que ce soit l'état $s_i$ qui ait émis le symbole de rang $t$ de $S_l$ et l'état $s_j$ qui ait émis celui de rang $t+1$.
- $\gamma_t^l(i)$ est la probabilité que le symbole de rang $t$ de la séquence $S_l$ soit émis par l'état $s_i$.

Ces deux quantités sont calculées par l'algorithme *Forward-Backward*.

Les algorithmes de classification par mélange de HMM visent à identifier de manière automatique la structure cohérente d'un groupe de séquences, de sorte que la proximité des observations d'une même classe $c$ vis-à-vis du modèle HMM de leur classe, définie par la fonction de vraisemblance $P(S_i|c_i = c, \phi_c)$, est similaire alors qu'elle est dissimilaire pour les observations relevant de classes différentes.

La classification automatique par mélange de HMM a été utilisée dans de nombreux contextes et par des chercheurs de différentes disciplines comme la *reconnaissance de parole* [Juang and Rabiner, 1985] et la *bioinformatique* [Durbin *et al.*, 1998]. En supposant qu'une séquence d'observation $S_i$ est produite selon une distribution d'un mélange de $k$ composantes, ces algorithmes utilisent le formalisme des *nuées dynamiques* pour estimer l'ensemble des paramètres des classes $\phi = \{\phi_1, \phi_2, \dots, \phi_k\}$ qui maximisent la fonction de vraisemblance suivante :

$$P(S|\phi) = \prod_{i=1}^{n} P(S_i|\phi_{c_i}) \quad (4.21)$$

L'algorithme des *nuées dynamiques* utilisé dans ce cas est itératif et a le déroulement suivant :

1. chaque séquence d'observation $S_i$ est affectée à la classe $c$ de paramètre $\phi_c$ tel que :

$$c = argmax_{1\leq c\leq k}\{P(S_i|\phi_c) = P(S_i|c_i = c, \phi_c)\} \quad (4.22)$$

2. Réajuster à l'aide de l'algorithme *Baum Welch* les paramètres des $k$ modèles HMM de classes $\phi_c = (A_c, B_c, \pi_c)$ $(c = 1, \dots, k)$ avec les $n_c$ séquences d'observation affectées à chaque classe $c$.
3. Recommencer en 1 jusqu'à ce que le critère de convergence soit rempli (*i.e.* qu'on ait atteint le maximum de vraisemblance).

Cette approche d'estimation des différents paramètres du modèle de mélange de HMM est effectuée d'une manière qui peut être qualifiée de « dure ». En effet, à chaque itération une séquence peut intervenir dans l'évaluation des paramètres d'une seule classe, celle à laquelle elle a été attribuée. Un inconvénient de cette approche est qu'elle peut échouer pour des jeux de données séquentielles où il n'y a pas de séparation nette entre les modèles de classes recherchées. Pour pallier à ce problème, certains auteurs ont proposé une version "*soft*" de l'algorithme d'estimation, et l'ont appliqué à la classification de séquences vidéo. Contrairement à la formulation des approches dites "*dures*", cette nouvelle proposition autorise chaque séquence à contribuer à l'estimation de plus d'un HMM par lequel elle peut être générée.

L'idée principale de cette proposition est de définir pour chaque séquence $S_l$ associée à un individu $i$ de la population un *vecteur d'adhésion* noté $z_l = \{z_{l,1}, z_{l,2}, \cdots, z_{l,k}\}$, où chaque $z_{l,c}$ possède une probabilité a priori $p_c$ pour qu'il soit égale à 1 (*i.e.* la séquence $S_l$ est générée par le $c^{ème}$ HMM). Dans ce cas, les paramètres à estimer s'étendent à l'ensemble $\phi = \{\phi_c, p_c | 1 \leq c \leq k\}$ qui maximise la fonction de vraisemblance modifiée suivante :

$$P(S|\phi) = \prod_{l=1}^{n}\sum_{c=1}^{k} P(S_l|\phi_c) * p_c \quad (4.23)$$

D'une manière analogue à l'algorithme des *nuées dynamiques*, l'algorithme proposé est itératif de type Espérance-Maximisation EM, et opère en deux étapes :

**Espérance**

Dans cette étape, l'algorithme calcule les probabilités a posteriori que $z_{l,c} = 1$ sachant que la séquence $S_l$ étant observée.

$$w_{l,c} = \frac{P(S_l|\phi_c) * p_c}{\sum_{u=1}^{k} P(S_i|\phi_u) * p_u} \qquad pour\ 1 \le c \le k \tag{4.24}$$

**Maximisation**

Comme pour l'algorithme EM déjà présenté, l'étape de maximisation sert à ajuster les différents paramètres des $k$ modèles de classe $\phi_c = \left(A^{(c)}, B^{(c)}, \pi^{(c)}\right)$, en pondérant cette fois chaque individu $l$ par sa probabilité a posteriori $w_{l,c}$.

$$p_c = \frac{\sum_{l=1}^{n} w_{l,c}}{n} \qquad pour\ 1 \le c \le k \tag{4.25}$$

Les équations 4.18, 4.19 et 4.20 sont ensuite modifiées comme suit :

$$\pi_i^{(c)} = \frac{\sum_{l=1}^{n} w_{l,c} * \gamma_1^l(i)}{\sum_{l=1}^{n} w_{l,c}} \qquad pour \begin{array}{l} 1 \le i \le R \\ 1 \le c \le k \end{array} \tag{4.26}$$

$$a_{i,j}^{(c)} = \frac{\sum_{l=1}^{n} w_{l,c} * \left(\sum_{t=1}^{T_l-1} \left(\xi_t^l(i,j)\right)\right)}{\sum_{l=1}^{n} w_{l,c} * \left(\sum_{t=1}^{T_l-1} \gamma_1^l(i)\right)} \qquad \begin{array}{r} 1 \le i \le R \\ pour\ 1 \le j \le R \\ 1 \le c \le k \end{array} \tag{4.27}$$

$$b_j^{(c)}(s) = \frac{\sum_{l=1}^{n} w_{l,c} * \left(\sum_{t=1 \ldots T_l-1 \text{ où } v_s = e_{l,t}} \gamma_1^l(j)\right)}{\sum_{l=1}^{n} w_{l,c} * \left(\sum_{t=1}^{T_l-1} \gamma_1^l(j)\right)} \qquad \begin{array}{r} 1 \le j \le R \\ pour\ 1 \le s \le m \\ 1 \le c \le k \end{array} \tag{4.28}$$

#### *4.2.2.3. Conclusion*

Contrairement aux approches de classification automatique fondées sur un indice de proximité, le principal intérêt des méthodes probabilistes de classification par modèles de mélange est *l'interprétabilité* des classes construites, étant donnée qu'elles travaillent sur les observations elles mêmes et non pas sur la proximité entre les séquences. Les modèles obtenus peuvent ainsi être rapidement exploités pour classer de nouvelles séquences et estimer leur évolution future. Cependant, l'algorithme d'évaluation des paramètres des modèles des classes souffre de certaines faiblesses que nous pouvons résumer dans les points suivants :

- Le nombre de classes $k$, pour les deux approches par mélange de chaînes de Markov et par modèles de Markov cachés, doit être fixé à l'avance,
- Pour les HMM, le nombre d'états cachés doit être également défini à l'avance,
- Les deux approches requièrent un partitionnement initial en $k$ classes des séquences comme première configuration de leurs processus d'apprentissage (EM pour les chaines de Markov et les *nuées dynamiques* pour les HMM). Cette étape d'initialisation se fait généralement de manière aléatoire et l'algorithme converge ensuite vers un maximum local. Par exemple, le maximum de la vraisemblance (équation 4.6), obtenu à la fin de la mise en œuvre de

l'algorithme EM avec un mélange de chaines de Markov, dépend du choix initial des probabilités conditionnelles d'appartenance aux classes de chaque individu $P(c_i = c|S_i, \phi)$. La vraisemblance maximale obtenue par cet algorithme n'est donc pas forcément le maximum global [Wu, 1983]. En conséquence, la phase d'initialisation est très importante, car une mauvaise initialisation pourrait conduire à un modèle non adéquat et la convergence de l'algorithme peut devenir très lente [Fraley and Raftery, 1998].

## 4.3. Autres approches de classification de données séquentielles

Afin de bénéficier des avantages des deux familles d'approches décrites précédemment (les approches probabilistes par modèles de mélange et les approches fondées sur la notion de proximité), Oates *et al.* [Oates *et al.*, 1999] ont proposé de combiner ces deux formalismes en y intégrant à la fois la *Dynamic Time Warping* et les *modèles de Markov cachés* dans un même cadre de classification automatique de données séquentielles. Dans cette nouvelle approche *mixte* (*hybride*), la DTW et les modèles HMM se complètent afin de résoudre les problèmes liés à chacune d'entre elles. Les auteurs considèrent ici le problème de classification d'un ensemble de $n$ séquences $S = \{S_1, S_2, \dots, S_n\}$ par un mélange de modèles de Markov cachés en supposant que la *Dynamic time Warping* associée à un algorithme de *classification ascendante hiérarchique* peut fournir un bon partitionnement initial de l'ensemble des séquences qui servira comme initialisation de l'algorithme d'estimation des paramètres des modèles.

Pour chaque classe retournée par l'algorithme hiérarchique associée avec la DTW, un modèle de Markov caché est construit en appliquant la procédure d'arrangement ci-dessus sur les séquences supposées appartenir à cette classe.

1: $S_0, S_0^{'} \leftarrow S$;
2: ***Répéter***
3: $S_0 \leftarrow S_0^{'}$;
4: *entraîner le HMM $\phi$ avec les séquences de $S_0$*;
5: $S_0^{'} \leftarrow$ *les séquences de $S_0$ acceptées par le HMM $\phi$*;
6: ***jusqu'***à$(S_0 = S_0^{'})$

Une séquence $O$ est dite acceptée par un HMM $\phi$ si elle vérifie l'hypothèse $P_\phi\left(log_{vraisemblance} \leq log\left(P(O|\phi)\right)\right) > seuil_fix$é, où $P_\phi$ est la distribution de probabilité empirique du logarithme de vraisemblance, calculée à partir d'un échantillon de séquences générées par ce modèle $\phi$.

Suite à l'application de cette procédure de réarrangement pour chaque classe de la partition initiale, on essaie d'affecter les séquences non acceptées par les HMM de leurs propres classes aux autres classes de la partition. Finalement, les séquences qui demeurent inadaptées à tous les HMM seront placées dans un même ensemble et ensuite classées en utilisant une approche de classification par mélange de HMM, basée sur la

procédure de réarrangement précédente. L'avantage de cette approche est qu'elle permet de fournir des solutions aux problèmes de sélection du nombre de classes et de partitionnement initial, tout en obtenant des classes significatives et facilement interprétables. L'inconvénient majeur réside dans la procédure de réarrangement utilisée qui peut retirer certaine bonne séquence de sa classe adéquate, en même temps que les séquences aberrantes.

Saunier et Sayed [Saunier and Sayed, 2006], ont proposé également une solution au problème lié à la connaissance préalable du nombre de classes pour une approche de classification par mélange de HMM. Cette proposition met en œuvre le formalisme des *nuées dynamiques* associé aux modèles de Markov cachés et une simple heuristique permettant de déterminer automatiquement le nombre de classes de la partition finale. Cet algorithme a été appliqué sur des séquences de conflit de trafic d'automobiles afin d'améliorer la sécurité routière.

Certains auteurs ont réfléchi à adapter les outils d'analyse de données symboliques pour le cas de données séquentielles et ainsi proposer une méthodologie ad-hoc. Dans ce cadre, Touati *et al.* [Touati *et al.*, 2006] ont proposé un cadre d'analyse des trajectoires-patients atteints d'un infarctus aigu du myocarde en utilisant *l'analyse de données symboliques*. L'objectif est de connaître les chances de survie d'un patient atteint d'un Infarctus aigu du myocarde dont on connaît la trajectoire hospitalière (succession des passages par les services de soins pendant un séjour d'hospitalisation). Pour cela, ils ont considéré la trajectoire hospitalière comme une séquence temporelle à description symbolique, qui peut facilement être exploitée par les méthodes de fouilles de données (les motifs fréquents, les arbres de décision, etc.). Ce cadre d'analyse consiste à mettre en place plusieurs outils qui partent de la *construction* de la trajectoire de soins (sous forme d'un objet symbolique), jusqu'à la *prédiction* de la survie du patient en passant par un module de *classification automatique*. En effet, une fois la trajectoire de soins construite, les auteurs ont proposé de :

- appliquer *l'algorithme des formes fortes* à l'ensemble des trajectoires-patients afin d'identifier des trajectoires-types. Une trajectoire-type est définie comme les trajectoires identiques réalisées par des patients ayant effectués des passages de soins dans le même ordre chronologique mais à des dates différentes.
- élaborer des classes de trajectoires pertinentes en utilisant l'approche SAPRIORI définie pour la découverte de motifs fréquents et propre aux données symboliques. L'analyse des classes obtenues par les experts médecins leur permet par la suite de concevoir une nouvelle répartition des trajectoires dans les classes, décrites sous forme de données symboliques.
- développer un module de prédiction en exploitant l'ensemble des trajectoires-types et des classes de trajectoires obtenues. Ce module utilise une méthode

descendante hiérarchique sous la forme d'un arbre de décision symbolique baptisée SYCLAD [Limam, 2005]. Cette dernière opère en segmentant la population des individus en deux classes, utilisant pour cela la variable descriptive la plus discriminante, de façon à avoir des sous-populations, appelées *nœuds*, contenant chacune le plus possible de patients d'une seule classe (Décès Oui/Non). Cette opération est réitérée sur chaque nouveau nœud obtenu jusqu'à ce que la séparation des individus ne soit plus possible ou plus souhaitable au sens d'un critère d'homogénéité $\alpha$. Les auteurs ont proposé d'implémenter cette approche de deux manières différentes : la première considère que les unités statistiques sont les patients décrits par leur trajectoire, alors que dans la deuxième, elles correspondent à l'ensemble des trajectoires-types décrites sous forme de données symboliques. L'analyse des deux arbres de décision obtenus a montré l'influence des trajectoires de soins pour la survie des patients atteint d'un infarctus aigu du myocarde.

# CADRE GENERIQUE D'ANALYSE DE DONNEES SEQUENTIELLES : APPLICATION AUX TRAJECTOIRES HOSPITALIERES


## Résumé

*Dans ce chapitre, nous proposons un nouveau cadre générique d'analyse de données séquentielles, basé sur le couplage entre l'approche de classification par b-coloration présentée dans le chapitre 3 et les chaines de Markov. Nous présentons ensuite l'application de la méthodologie au domaine médical. Nous utilisons pour cela des jeux de trajectoires hospitalières extraits d'une base de données PMSI-2003 fournie par l'Agence Régionale d'Hospitalisation Rhône-Alpes (ARH-RA) et qui concerne l'ensemble des établissements de santé de la région (publics et privés). Outre l'évaluation de la pertinence de l'approche proposée, cette application médicale permet de développer de nouvelles perspectives pour l'aide à la décision hospitalière. C'est pourquoi dans une dernière étape, nous présentons l'outil d'aide à la décision hospitalière que nous avons mis en place pour concrétiser, sur un plan technique, nos différentes contributions théoriques.*


## Sommaire

# Chapitre 5

# Cadre générique d'analyse de données séquentielles : application aux trajectoires hospitalières

*" Le temps découvre les secrets ; le temps fait naître les occasions ; le temps confirme les bons conseils. "*

Jacques-Bénigne Bossuet, " Sermons"

## 5.1. Introduction

Dans ce chapitre, nous proposons un cadre générique d'analyse de données séquentielles basé sur une méthodologie hybride combinant notre approche de classification par *b-coloration de graphes* présentée dans le chapitre 3 et les chaines de Markov. Cette méthodologie fournit pour chacune des classes de séquences obtenues une double description : premièrement, notre approche délivre un ensemble de *séquences types* (*profils*) qui sont le reflet des propriétés de leurs classes, et qui garantissent également une séparation nette de celles-ci vis-à-vis des autres classes de la partition. D'autre part, elle permet de faire correspondre à chaque classe un *modèle probabiliste de génération de données* (chaine de Markov) résumant les relations entre les différents états des séquences de la classe. Ce modèle assure donc une meilleure *interprétabilité* des classes construites et peut être appliqué pour classer de nouvelles séquences (*classement*) et estimer leurs évolutions futures (*prévision*).

Dans la suite de ce chapitre, nous proposons d'appliquer notre méthodologie d'analyse de données séquentielles. Pour cela, nous partons d'un jeu de données médicales relatives au système PMSI. Dans ce jeu de données, nous disposons d'un ensemble de trajectoires hospitalières de patients de la région Rhône-Alpes où (1) une trajectoire $S_i = \{e_{i,1}, e_{i,2}, \dots, e_{i,T_i}\}$ est définie comme l'ensemble des $T_i$ séjours hospitaliers $e_{i,j}$ effectués successivement par un patient $i$, (2) un séjour $e_{i,j}$ étant caractérisé par l'ensemble de renseignements PMSI, à savoir des informations générales sur le patient (sexe, âge) et des informations concernant ses soins (la classe de son séjour, différents

diagnostics relatifs à la pathologie, l'ensemble des actes médicaux réalisés pendant le séjour, le mode et le mois de sortie, la durée de séjour, etc.).

D'autre part, nous avons mis en œuvre notre proposition au sein d'une plateforme logicielle appelée *"Analyse de trajectoires PMSI".* Il s'agit d'une application dédiée à l'analyse des trajectoires hospitalières du PMSI et à l'implémentation de nos travaux sur le couplage entre l'approche de *classification par la b-coloration de graphes* et les *chaînes de Markov*. L'approche par *b-coloration* fournit des classes de trajectoires homogènes caractérisées chacune par un ensemble de *trajectoires types* (*profils de patient*), alors que les chaînes de Markov permettent d'interpréter les classes au moyen de modèles probabilistes formant un *cadre automatique de prévision* des trajectoires de soins. En effet, pour un patient ayant eu une suite de séjours hospitaliers, il s'agit dans un premier temps d'identifier la classe de trajectoires dont il se rapproche le plus. Dans un deuxième temps, si c'est nécessaire, nous pouvons prévoir quel sera le séjour suivant le plus probable, et d'en estimer les caractéristiques principales (type de séjour -classe-, diagnostic médical principal, mode de sortie, actes à subir, etc.). Chaque propriété est affectée des probabilités obtenues d'après le modèle de Markov établi pour la classe de trajectoires.

Ce chapitre est organisé de la façon suivante. Dans la section 5.2, nous développons notre méthodologie pour l'analyse de données séquentielles. L'application de la méthodologie à un jeu de données réelles, utilisant les trajectoires médicales du PMSI, est fournie dans la section 5.3 ce qui débouche sur une première discussion à partir des résultats obtenus. La section 5.4 présente l'architecture générale de l'outil logiciel d'aide à la décision pour l'analyse des trajectoires-patients qui a été développé en Java/XML. Enfin, la section 5.5 conclut le chapitre et expose les perspectives qui ont été d'ores et déjà envisagées pour la méthodologie.

## 5.2. Cadre d'analyse de données séquentielles

Dans ce chapitre, nous continuons dans l'idée de coupler les approches probabilistes de classification aux approches fondées sur la notion de proximité car c'est une solution qui semble répondre aux attentes de l'analyse des données séquentielles. Dans cette optique, un cadre d'analyse de données séquentielles a été élaboré qui exploite les avantages des deux approches : la classification par *b-coloration de graphes* décrite dans le troisième chapitre et le modèle de mélange markovien pour pouvoir facilement interpréter les classes, identifier la classe d'appartenance des nouvelles séquences et prévoir l'évolution de séquences déterminées. Nous avons cherché à proposer une méthode qui répond aux inconvénients des approches existantes. Ainsi, la méthode ne nécessite pas de connaître à l'avance le nombre de classes caractérisant la population étudiée. La méthodologie s'applique aux séquences de longueur quelconque dont les états sont décrits par des variables hétérogènes, et elle utilise la distance d'édition modifiée pour mieux prendre en

compte la présence d'états communs entre des séquences de taille différentes. Les sections suivantes décrivent l'approche dans le détail [Elghazel *et al.*, 2007b].

Supposons l'ensemble de séquences $S = \{S_1, S_2, \dots, S_n\}$, qui peuvent être de longueurs différentes, où chaque séquence $S_i = \{e_{i,1}, e_{i,2}, \dots, e_{i,T_i}\}$ est donnée par une suite de $T_i$ états $e_{i,j}$ observés successivement pour un individu $i$ et où chaque état $e_{i,j}$ est décrit sur un ensemble de $p$ variables hétérogènes $Y = \{Y_1, Y_2, \dots, Y_p\}$. Le processus de classification considéré dans ce chapitre vise à structurer les séquences contenues dans $S = \{S_1, S_2, \dots, S_n\}$ en fonction de leurs ressemblances, sous la forme d'un ensemble de classes *homogènes*, *significatives* et *contrastées*. Dans ce cas de séquences complexes, il est nécessaire de les structurer et de les homogénéiser pour pouvoir les exploiter à des fins décisionnelles, ce que vise notre cadre d'analyse. C'est ainsi que nous considérerons dans la suite que chaque état $e_{i,j}$ n'est plus décrit par un ensemble de $p$ variables, mais par un groupe $g_{i,j}$ obtenu par classification des $p$ variables descriptives. La séquence construite devient ainsi $S_i = \{g_{i,1}, g_{i,2}, \dots, g_{i,T_i}\}$.

D'une manière similaire au problème de classification traité dans le troisième chapitre, la classification des données séquentielles est aussi considérée comme un problème de partitionnement de graphes. Pour cela, nous considérons la représentation topologique de l'ensemble des séquences à grouper $S = \{S_1, S_2, \dots, S_n\}$ par un graphe *complet*, *non orienté* et *pondéré* $G = (\boldsymbol{V}, \boldsymbol{E})$ pour lequel les sommets $\{v_1, v_2, \dots, v_n\}$ sont les séquences à classer (le sommet $v_i$ correspond à la séquence $S_i$) et les arêtes les liens pondérés par les dissimilarités entre les paires de séquences. Le graphe $G$ est traditionnellement représenté par un tableau de dissimilarités symétrique $D = \{d(S_i, S_j) | S_i, S_j \in S\}$ de taille $nxn$.

### 5.2.1. Construction de la matrice des dissimilarités entre séquences

Compte tenu de la nature qualitative des états $g_{i,j}$ des séquences $S_i$ à analyser, nous avons choisi d'utiliser la *distance d'édition* présentée précédemment pour évaluer la ressemblance entre les séquences temporelles. Comme précisé au chapitre précédent, le problème de l'évaluation de la distance d'édition entre deux séquences temporelles (considérées comme deux chaînes de caractères) est une généralisation du problème de l'évaluation de la longueur d'une plus longue sous-séquence commune à ces deux séquences. La distance d'édition est donnée dans ce cas par l'équation 4.2. Cependant, cette formulation souffre de plusieurs inconvénients dus à la différence des tailles des séquences traitées, ce qui rend son application directe non appropriée à notre problématique de classification. Les trois séquences $X = (a, b, a, b)$; $Y = (c, d, c, d)$ et $Z = (a, b, e, f, g, h, i, j)$ fournissent par exemple une bonne illustration de cette faiblesse. En effet, la distance entre $X$ et $Z$ (avec deux symboles en commun : $a$ et $b$ en commun) devrait être plus petite que celle entre $X$ et $Y$ (qui n'ont aucun symbole en commun). Toutefois, on

calcule que la distance d'édition (équation 4.2) prend la valeur 8 pour $X$ et $Y$ $(4 + 4 - 2 * 0 = 8)$ et également comme pour $X$ et $Z$ $(4 + 8 - 2 * 2 = 8)$.

Cette limitation nous a poussés à modifier la distance d'édition afin qu'elle prenne mieux en compte la ressemblance entre les séquences. La *distance d'édition modifiée* entre deux séquences $S_i$ et $S_j$ est définie ainsi par le rapport de la distance d'édition originale sur la somme des longueurs des deux séquences (voir équation 5.1). Les valeurs et maximum de cette mesure sont ainsi respectivement 0.0 et 1.0.

$$d_{E\,m}(S_i, S_j) = \frac{|S_i| + |S_j| - 2 * LCS(S_i, S_j)}{|S_i| + |S_j|} \tag{5.1}$$

## 5.2.2. Classification des séquences

Les séquences à analyser sont maintenant associées à une matrice de dissimilarités symétrique $D = \{d(S_i, S_j) | S_i, S_j \in S\}$ de taille $nxn$. Il s'agit dans cette partie de définir une méthodologie pour la classification automatique de l'ensemble de séquences $S = \{S_1, S_2, \dots, S_n\}$. L'objectif est de définir et de construire une typologie de séquences en classes *homogènes* et *bien séparées*, mais aussi de résumer l'information qu'elles contiennent dans des modèles en vue de les interpréter et les appliquer par la suite à des fins de *classement* et de *prévision*. Pour ce faire, nous avons proposé la démarche suivante :

### *a. Classification automatique des séquences par l'approche de b-coloration*

Il s'agit ici d'utiliser l'approche présentée dans le troisième chapitre. Cette approche vise à structurer les séquences contenues dans $S = \{S_1, S_2, \dots, S_n\}$ en fonction de leurs ressemblances, sous la forme d'un ensemble de classes *homogènes* et *bien distinctes* caractérisées chacune par un ensemble de *séquences types* (*dominantes*). Celles-ci sont le reflet des propriétés de leur classe mais garantissent aussi une séparation nette de celle-ci vis-à-vis des autres classes de la partition.

### *b. Modèles de mélange et b-coloration.*

Compte tenu des performances significatives des chaînes de Markov dans l'élaboration de modèles probabilistes de génération de données résumant les relations entre les états des séquences traitées, et vues les difficultés des approches de classification automatique par modèles de mélange à établir les probabilités initiales et le nombre de classes, nous avons proposé utiliser les résultats de classification obtenus dans l'étape 1, par l'approche de *b-coloration*, comme alternative au problème d'initialisation des paramètres des modèles à estimer. L'initialisation se fait ainsi à partir de la classification obtenue par la b-coloration pour distribuer les individus dans les classes. Les probabilités initiales $P(c_i = c|S_i, \phi)$ sont donc égales à 1 pour la classe d'appartenance donnée par la *b-coloration*, et nulles pour toutes les autres classes de la partition.

D'autre part, afin d'estimer les paramètres du modèle $\phi_c = (\pi_c, A_c)$ de chaque classe $c$ de la partition $(c = 1, \dots, k)$, l'algorithme *Espérance-Maximisation* (EM) est appliqué uniquement sur les séquences *non critiques*. Dans notre méthodologie, une séquence est dite *critique* si elle est *dominante* (son sommet correspondant dans le *graphe seuil optimal* est *dominant* de sa couleur, ce qui signifie que toute modification de cet élément risque de perturber fortement la représentativité de sa classe) ou si elle est le *seul support de la dominance* d'une séquence dominante voisine (son sommet correspondant dans le graphe seuil optimal est le seul voisin dans sa couleur à un sommet dominant d'une autre couleur de ce graphe). Ainsi, dans l'objectif de maintenir la propriété de dominance de la *b-coloration* et donc la stabilité de la partition obtenue, nous considérons qu'une telle séquence ne doit pas changer sa classe initiale $c(i)$ tout au long du processus d'estimation des paramètres des modèles de classes. Pour ce faire, nous proposons de conserver les mêmes probabilités conditionnelles d'appartenance aux classes $P(c_i = c|S_i, \phi)$ pour les séquences critiques à chaque étape *Espérance* de l'algorithme itératif EM. Cette stratégie garantit ainsi d'avoir, à la fin du déroulement de l'algorithme, des séquences dominantes dans chaque classe et permet aussi aux séquences *non critiques* d'intervenir dans l'estimation des paramètres des classes susceptibles de la reproduire.

La nouvelle étape *Espérance* du processus d'estimation est la suivante :

**pour** $i$ *de* 1 à $n$ **faire**
  **si** $S_i$ *est une séquence critique* **alors**
    $P(c_i = c(i)|S_i, \phi) = 1$;
    **pour** $c$ *de* 1 à $k$ **tel que** $c \neq c(i)$ **faire**
      $P(c_i = c|S_i, \phi) = 0$
    **fin pour**
  **sinon**
    **pour** $c$ *de* 1 à $k$ **faire**
      $P(c_i = c|S_i, \phi) = \frac{P(S_i|c_i=c,\phi_c)*P(c)}{\sum_{u=1}^{k} P(S_i|c_i=u,\phi_u)*P(u)}$
    **fin pour**
  **fin si**
**fin pour**

#### c. Prévision de la suite de séquence

Une fois le modèle de mélange appris sur l'ensemble des séquences temporelles $S = \{S_1, S_2, \dots, S_n\}$, nous pouvons nous servir pour faire de la prévision *on-line* de la suite d'une séquence $S_a$ (nouvelle ou existante) après avoir observer son historique $\{g_{a,1}, g_{a,2}, \dots, g_{a,T_a}\}$ ($g_{a,j}$ est le groupe associé à l'état $e_{a,j}$). Il s'agit ainsi de :

1. Affecter cette séquence à la classe $c_a$ la plus susceptible de la reproduire (la classe pour la quelle $S_a$ a la plus grande probabilité d'appartenance $P(S_a|c_a = c, \phi_a)$) : *propriété en ligne de classification*.

$$c_a = argmax_{1 \leq c \leq k}\{P(S_a|\phi_c) = P(S_a|c_a = c, \phi_c)\} \quad (5.2)$$

Une fois la classe de $S_a$ choisie, nous pouvons également imaginer présenter les *séquences dominantes* (*séquences-types*) de cette classe à l'utilisateur. Celles-ci lui fournissent une idée plus claire sur le profil de l'individu $a$ en question et les propriétés de sa classe $c_a$ choisie.

2. Utiliser $A_{c_a}$ , la matrice $m * m$ ($m$ est le nombre d'états possibles) de transitions, associée à la chaine de Markov $\phi_{c_a}$ de la classe $c_a$, afin de prévoir la suite de la séquence temporelle $S_a$.

$$g_{a,T_a+1} = argmax_{1 \leq z \leq m}\{a_{c_a}(e_{a,T_a}, z)\} \quad (5.3)$$

## 5.3. Application aux trajectoires hospitalières du PMSI

### 5.3.1. Description et préparation des données

Nous présentons, dans cette section, les résultats de notre cadre d'analyse appliqué aux données séquentielles relatives au *Programme de Médicalisation des Systèmes d'Information* français. Il s'agit d'un jeu de trajectoires médicales issues du système PMSI-2003, fourni par l'Agence Régionale d'Hospitalisation Rhône-Alpes (ARH-RA) et recensant les données PMSI de tous les établissements de santé de la région (publics et privés). Ce jeu de données séquentielles comprend l'ensemble des séjours hospitaliers effectués sur une année pour tous les patients de la région Rhône-Alpes. Afin de pouvoir retracer la trajectoire d'un patient sur une période donnée, l'Agence Régionale d'Hospitalisation relie les séjours grâce à un numéro de chaînage anonyme, identifiant régional unique du patient défini à partir de ses informations personnelles (sexe, date de naissance, numéro de sécurité sociale).

Grâce au PMSI, il est possible d'avoir pour chacun des séjours un ensemble de caractéristiques *statiques* et *dynamiques.* Il s'agit d'informations systématiquement renseignées, concernant le patient (sexe, âge) et les soins prodigués au sein de l'établissement (diagnostic principal relatif à la pathologie, diagnostics associés, ensemble d'actes médicaux réalisés pendant le séjour, mode et mois de sortie, durée de séjour, etc.), auxquelles s'ajoute le groupe homogène de malades (GHM) du séjour (voir figure 5.1).

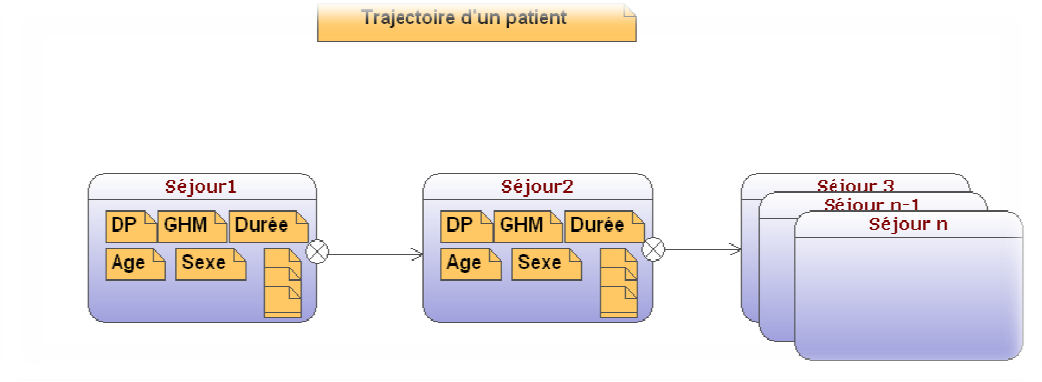



**Figure 5.1- Représentation d'une trajectoire hospitalière d'un patient**

Pour exploiter ces trajectoires et compte tenu de la nature *hétérogène* et *complexe* des informations (à la fois classique et symbolique) caractérisant leurs états-séjours, il est nécessaire de *réduire* ces informations et de les *structurer*. Un partitionnement de ces états-séjours par l'approche de classification par *b-coloration de graphes* présentée dans le troisième chapitre s'avère une solution appropriée à cette phase préparatoire des trajectoires de soins. Cette méthode permet de construire une partition fine de l'ensemble de séjours en classes *homogènes* et *bien séparées*. Les séjours seront alors associés à une classe construite sur la base des *variables médico-économiques* mentionnées précédemment.

Afin de faciliter les tâches d'interprétation et de validation des résultats, dans une étape ultérieure, par nos experts médecins, nous nous sommes contentés, dans cette phase de préparation primordiale, d'utiliser les classes de séjours (GHM : Groupes Homogènes de Malades) déjà présentes dans les informations fournies par le système PMSI. Notons aussi que celles-ci ont été construites par un arbre de décision portant sur les valeurs des variables *médico-économiques* décrivant les séjours des patients.

D'autre part, pour mieux expliquer les dimensions médicale et économique des états-séjours, il nous a été proposé, par nos experts médecins, de croiser la variable « *groupe de séjour* » (GHM) avec la variable « *diagnostic principal* » (DP) dans la description des états de la séquence de soins. En effet, si le GHM représente plus l'aspect économique du séjour, le DP fournit quant à lui une meilleure idée de l'état médical du patient. Dans le tableau 5.1 suivant, nous présentons l'exemple de quelques trajectoires hospitalières.

| $\boldsymbol{Trajectoire\ (S_i)}$ | $\boldsymbol{Etat\ 1}$ | $\boldsymbol{Etat\ 2}$ | $\boldsymbol{Etat\ 3}$ | $\boldsymbol{Etat\ 4}$ |
|---|---|---|---|---|
| $\boldsymbol{S_1}$ | DRG12-DP1 | DRG7-DP6 | DRG7-DP5 | DRG13-DP3 |
| $\boldsymbol{S_2}$ | DRG8-DP2 | DRG11-DP33 | DRG12-DP1 | |
| $\boldsymbol{S_3}$ | DRG1-DP21 | DRG1-DP21 | DRG7-DP6 | DRG23-DP2 |

**Tableau 5.1- Exemples de trajectoires hospitalières**

Dans l'objectif de conserver la particularité de chacune de ces deux informations (GHM et DP) présentes dans les trajectoires, nous avons proposés de les considérer séparément dans le calcul de ressemblance entre les trajectoires, comme le montre le tableau suivant :

| $\boldsymbol{Trajectoire\ (S_i)}$ | $\boldsymbol{S_{i,k}}$ | $\boldsymbol{Etat\ 1}$ | $\boldsymbol{Etat\ 2}$ | $\boldsymbol{Etat\ 3}$ | $\boldsymbol{Etat\ 4}$ |
|---|---|---|---|---|---|
| $\boldsymbol{S_1}$ | Série GHM : $(\boldsymbol{S_{1,1}})$ | DRG12 | DRG7 | DRG7 | DRG13 |
| | Série DP : $(\boldsymbol{S_{1,2}})$ | DP1 | DP6 | DP5 | DP3 |
| $\boldsymbol{S_2}$ | Série GHM : $(\boldsymbol{S_{2,1}})$ | DRG8 | DRG11 | DRG12 | |
| | Série DP : $(\boldsymbol{S_{2,2}})$ | DP2 | DP33 | DP1 | |
| $\boldsymbol{S_3}$ | Série GHM : $(\boldsymbol{S_{3,1}})$ | DRG1 | DRG1 | DRG7 | DRG23 |
| | Série DP : $(\boldsymbol{S_{3,2}})$ | DP21 | DP21 | DP6 | DP2 |

**Tableau 5.2 - Description des états des trajectoires hospitalières**

A partir de ce tableau 5.2, la mesure de ressemblance entre deux séquences $S_i$ et $S_j$ que proposée dans ce cas, est donnée par l'agrégation des distances d'édition obtenues sur chaque série d'attributs (GHM et DP). Pour ce faire, la distance euclidienne est utilisée comme suit :

$$d(S_i, S_j) = \sqrt{\sum_{h=1}^{2} \left(d_{Em}(S_{i,h}, S_{j,h})\right)^2} \tag{5.4}$$

où $d_{Em}(S_{i,h}, S_{j,h})$ $(h \in \{1,2\})$ est la distance d'édition modifiée entre les deux séries d'attributs $S_{i,h}$ et $S_{j,h}$ ($h = 1$ pour *GHM* et $h = 2$ pour *DP*) correspondants respectivement aux trajectoires $S_i$ et $S_j$.

### 5.3.2. Expérimentations et performances

Afin d'évaluer les performances du cadre d'analyse proposé, nous avons mené un ensemble d'expérimentations sur deux échantillons de trajectoires hospitalières tirés de la base PMSI-2003 [Elghazel *et al.*, 2007b]. Les deux échantillons de tailles respectives 406 et 2050 trajectoires ont été construits par une procédure d'échantillonnage stratifié effectué sur la population globale (la totalité de la base). Les deux échantillons ont été élaborés de façon différente : nous avons sélectionné dans le premier échantillon les trajectoires de façon à avoir le minimum de groupes homogènes de malades et de diagnostics principaux (les plus fréquents dans la base totale), soient 222 GHM et 420 DP pour un échantillon stratifié de 406 trajectoires. Le deuxième échantillon comporte, quant à lui, des trajectoires avec une plus grande diversité de maladies (399 GHM et 1323 DP). Notons également que la taille des trajectoires dans les deux échantillons varie entre 4 et 30 séjours hospitaliers pour un même patient (certains séjours correspondent à des séances effectuées dans le cadre d'un même traitement, par ex. la chimiothérapie).

La qualité des classes de trajectoires obtenues par notre méthodologie a été comparée à celle des classes identifiées par un modèle de mélange de chaines de Markov introduit dans le chapitre précédent [Cadez *et al.*, 2000a]. Nous avons considéré, pour ce modèle, le même nombre $k$ de classes que celui retourné par l'approche de classification par *b-coloration*. D'autre part, une distribution aléatoire des trajectoires dans les $k$ classes a été utilisée pour initialiser les probabilités conditionnelles d'appartenance $P(c_i = c|S_i, \phi)$, nécessaires pour la mise en œuvre de l'algorithme d'estimation EM.

Sachant que le but de ces deux approches pour l'analyse de données séquentielles est de fournir des résultats facilement interprétables et exploitables par l'utilisateur humain, nous avons cherché à utiliser quelques mesures objectives pour une meilleure évaluation des résultats obtenus. Les deux indices de qualité suivants sont alors considérés :

### *a. Indice de performance de prévision (PP)*

Cet indice a été utilisé pour examiner le taux de *bonne prévision* de l'évolution des séquences $S_i = (g_{i,1}, g_{i,2}, \dots, g_{i,T_i})$ $\left(g_{i,j} = (GHM_{i,j}, DP_{i,j})\right)$ dans un jeu de données séquentielles $X$. L'idée principale de ce processus d'évaluation consiste à sélectionner les séquences $S_i$ séparément et à :

- Eliminer le dernier état $g_{i,T_i}$ de la séquence $S_i$.
- Classer la nouvelle séquence tronquée (que nous appelons $S_{tronquée}$) dans l'une $k$ classes existantes, en utilisant la formule 5.5. La classe choisie sera notée $c_i$.

$$c_i = argmax_{1\le c\le k}\{P(S_{tronquée}|c_i = c, \phi_c)\} \tag{5.5}$$

- Prévoir l'état $z$ le plus probable d'apparaître à la fin de cette séquence tronquée. Ceci est réalisé en utilisant la matrice de transition $A_{c_i}$ associée à la classe $c_i$ choisie. Cet état $z$ sera par la suite comparé à l'état réel $g_{i,T_i}$qui a été supprimé de la séquence comme suit :

$$PP_X = \frac{\sum_{i\in|X|}\omega_i}{|X|}$$
$$\text{où } \omega_i = \begin{cases} 1; & si\ g_{i,T_i} = argmax_{1\le z\le m}\{a_{c_i}(g_{i,T_i-1}, z)\} \\ 0; & sinon \end{cases} \tag{5.6}$$

Les performances de prévision ont été examinées pour les deux échantillons de trajectoires hospitalières. Les résultats sont obtenus à l'aide d'un processus de validation croisée. En effet, chaque jeu de donnés séquentielles (trajectoires hospitalières) a été divisé en 5 parties disjointes. Quatre de ces parties (80% des trajectoires) ont servi comme une base d'apprentissage et le reste (20%) pour la phase de test. Ce processus a été répété cinq fois et nous avons ensuite établi la moyenne des résultats. L'échantillon d'apprentissage sert à générer une typologie des trajectoires et à modéliser les classes obtenues à partir des deux cadre approches (la nôtre et le modèle de mélange markovien).

L'échantillon de test est ensuite utilisé pour évaluer les résultats de prévision. Pour une meilleure estimation des résultats obtenus, les performances de prévision ont été aussi évaluées pour la base d'apprentissage.

### *b. Indice d'homogénéité Intraclasse (HI)*

Cet indice est fondamental pour la validation d'un problème de classification automatique de données séquentielles. Proposée par Estasio *et al.* dans [Estacio-Moreno *et al.*, 2005], *l'homogénéité intraclasse* est considérée comme un indice probabiliste qui reflète la stabilité, la cohésion et la facilité d'interprétation des classes obtenues par un processus de classification automatique. Plus grande est la valeur d'*homogénéité intraclasse*, plus les classes de la partition sont compactes et facilement interprétables, ainsi que les concepts qu'elles représentent.

Pour une partition $P$ en $k$ classes $\{C_1, C_2, \dots, C_k\}$ de l'ensemble des séquences $S = \{S_1, S_2, \dots, S_n\}$, *l'homogénéité intraclasse* $HI(P)$ est définie par la moyenne des *homogénéités intraclasse* des $k$ classes de la partition $P$ comme suit :

$$HI(P) = \frac{\sum_{c=1}^{k} HI_c}{k} \tag{5.7}$$

*où :*

$$HI_c = \sum_{S_i \in c} \delta_i \tag{5.8}$$

*où*

- $\begin{cases} \delta_i = 0 \text{ si } P(c_i = c|S_i, \Phi) < 0{,}5 \\ \delta_i = 1 \text{ si } P(c_i = c|S_i, \Phi) \geq 0{,}5 \end{cases}$
- $P(c_i = c|S_i, \Phi)$ *est donnée par l'équation* 4.11
- $c_i$ *est la classe de la séquence* $S_i$
- $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_k\}$ *représente les paramères de tous les clusters*

Les tableaux 5.3 et 5.4 montrent les résultats obtenus sur les deux échantillons de trajectoires PMSI en terme de *performance de prévision* et *d'homogénéité intraclasse*. Ces valeurs indiquent clairement que notre cadre d'analyse permet de fournir les meilleures performances que le modèle de mélange de chaines de Markov.

L'efficacité de nos modèles décisionnels de classification automatique et de prévision a été confirmée. En effet, notre cadre d'analyse permet de fournir une typologie de trajectoires avec des classes homogènes, bien séparées et facilement interprétables pour un processus de prévision. On constate notamment que l'homogénéité intraclasse obtenue par notre modèle s'élève à 94% (contre 43% pour le *modèle de mélange markovien*) sur le

premier échantillon, alors qu'elle vaut 98% (contre 73% pour le *modèle de mélange markovien*) avec le deuxième échantillon.

| *Cadre d'analyse* | *HI* | *PP (Base d'apprentissage)* | *PP (Base de test)* |
|---|---|---|---|
| *b-coloration+chaîne de Markov* | *0.94* | *76,4%* | *58,5%* |
| *Modèle de Mélange Markovien* | *0.43* | *61,2%* | *45,1%* |

**Tableau 5.3 - Performances sur la première base (406 trajectoires)**

| *Cadre d'analyse* | *HI* | *PP (Base d'apprentissage)* | *PP (Base de test)* |
|---|---|---|---|
| *b-coloration+chaîne de Markov* | *0.98* | *86,48%* | *68,4%* |
| *Modèle de Mélange Markovien* | *0.73* | *75,8%* | *49,3%* |

**Tableau 5.4 - Performances sur la seconde base (2050 trajectoires)**

Toutefois, nous avons remarqué, dans notre cas, que l'algorithme EM converge au bout d'une seule itération avec les deux échantillons de trajectoires hospitalières. Ceci est certainement dû au faible nombre de séquences *non critiques* et à la forte *stabilité* des classes obtenues par l'approche de classification par *b-coloration*. Cette stabilité a été confirmée par l'examen du nombre de *séquences dominantes* (*types*) pour chaque classe. Le pourcentage de ces séquences-prototypes s'élève à 78,44% (*resp.* 85,56%) pour la partition des 406 (*resp.* 2050) trajectoires hospitalières. Le nombre important de ces séquences, dans un tel cas, garantit une séparation nette entre les classes de la partition et minimisent le nombre de séquences susceptibles de changer de classe (*séquences non critiques*), ce qui réduit l'impact de l'algorithme EM (section 5.2.2) pour le cas traité. Le modèle est ici quasiment équivalent à la modélisation de chaque classe, obtenue par l'approche de *b-coloration,* à l'aide d'une chaine de Markov. Néanmoins on peut envisager que pour toute autre application, l'approche basée sur l'algorithme EM permettrait de progressivement stabiliser la partition initialement obtenue par la b-coloration.

Les expérimentations réalisées sur les deux jeux de trajectoires PMSI-2003 ont montré l'utilité de notre cadre d'analyse comme un outil d'aide à la décision hospitalière qui permet de répondre aux deux problèmes de classification et de prévision des trajectoires de patients. En utilisant un tel système, les établissements de soins peuvent prévoir les modèles de visites (profils) futures qui leur permettront de mieux organiser leurs ressources (humaines et matérielles) et éventuellement d'évaluer les coûts de séjours dès l'arrivée des patients.

Suite à la sollicitation de nos partenaires hospitaliers, nous avons ensuite pensé affiner notre cadre d'analyse pour étoffer l'aide à la décision concernant la prévision des séjours patients. En effet, une fois l'état $g_{a,T_a+1}$ $(GHM, DP)$, identifié comme le plus probable à la suite de la séquence $S_a$, le système peut passer à un niveau d'estimation plus détaillée en évaluant certaines variables descriptives du nouvel état $g_{a,T_a+1}$. Ainsi pour notre

application, il est possible d'estimer (avec une certaine probabilité) les actes médicaux $Actes_a$ que peut subir le patient $a$ lors de son futur passage par l'hôpital. $Actes_a$ est donné par l'ensemble des traitements ayant une certaine probabilité d'apparition, à l'issue d'une transition de soins $g_{a,T_a} \rightarrow g_{a,T_a+1}$ dans la classe $c_a$, supérieure à un seuil donné $\alpha$.

$$Actes_a = \left\{t; Prob_{c_a/g_{a,T_a} \rightarrow g_{a,T_a+1}}(t) > \alpha\right\} \quad (5.9)$$

où $Prob_{c_a/g_{a,T_a} \rightarrow g_{a,T_a+1}}(t)$ est la probabilité d'avoir le traitement médical $t$ suite à une transition de soins $s_{a,T_a} \rightarrow s_{a,T_a+1}$ au sein de la classe $c_a$ (groupe de trajectoires hospitalières données).

## 5.4. Plateforme logicielle (*Analyse de trajectoires PMSI*)

Pour valider nos propositions, nous avons implémenté notre cadre d'analyse de données séquentielles sur une plateforme logicielle appelée *"Analyse de trajectoires PMSI"*. Il s'agit d'une application dédiée à l'analyse des trajectoires hospitalières du PMSI et à l'implémentation de nos travaux sur le couplage de l'approche de classification par la *b-coloration de graphes* et les *chaînes de Markov*.

Afin d'assurer une interaction simple et efficace avec les utilisateurs médecins à travers une interface conviviale et dans le but de les impliquer le moins possible dans le processus d'analyse des trajectoires, nous avons développé *"Analyse de trajectoires PMSI"* dans un environnement Java qui permet de gérer les droits d'accès selon le type de l'utilisateur. Nous distinguons alors deux types d'utilisateurs :

- ***Administrateur :*** il bénéficie de plusieurs privilèges par rapport aux autres utilisateurs. Il peut ainsi modifier les échantillons de données, gérer les utilisateurs du système. Au niveau applicatif, c'est lui qui va déclencher les opérations de préparation des données, de classification automatique et de modélisation des classes de trajectoires.
- ***Médecin :*** il possède les droits d'ajout, de modification, de recherche de trajectoires et peut lancer la prévision pour une trajectoire donnée. Mais il est dépourvu des privilèges de l'administrateur.

Nous présentons dans la suite l'architecture générale de la plateforme *"Analyse de trajectoires PMSI"* en détaillant les modules dédiés aux propositions faites dans le cadre de l'analyse des données séquentielles du PMSI. En effet, nous distinguons trois types de modules : le module d'exploration des trajectoires de soins, le module de préparation et d'analyse et le module d'aide à la décision. La plateforme est également ouverte à l'ajout d'autres modules que nous avons prévus d'intégrer dans la suite de nos travaux.

### 5.4.1. Le module d'exploration

Le module d'exploration de la plateforme *"Analyse de trajectoires PMSI"* permet de visualiser les différentes trajectoires de patients (disponibles dans la base) en analysant les informations associées à leurs épisodes de soins (voir figure 5.2 où deux trajectoires successives sont représentées avec deux couleurs différents pour faciliter la visualisation). Ce module permet également à l'utilisateur de déclencher les opérations classiques d'ajout et de suppression des séjours (pour des patients ayant déjà des trajectoires disponibles dans la base) ainsi que des trajectoires (pour des nouveaux patients arrivant dans le système).

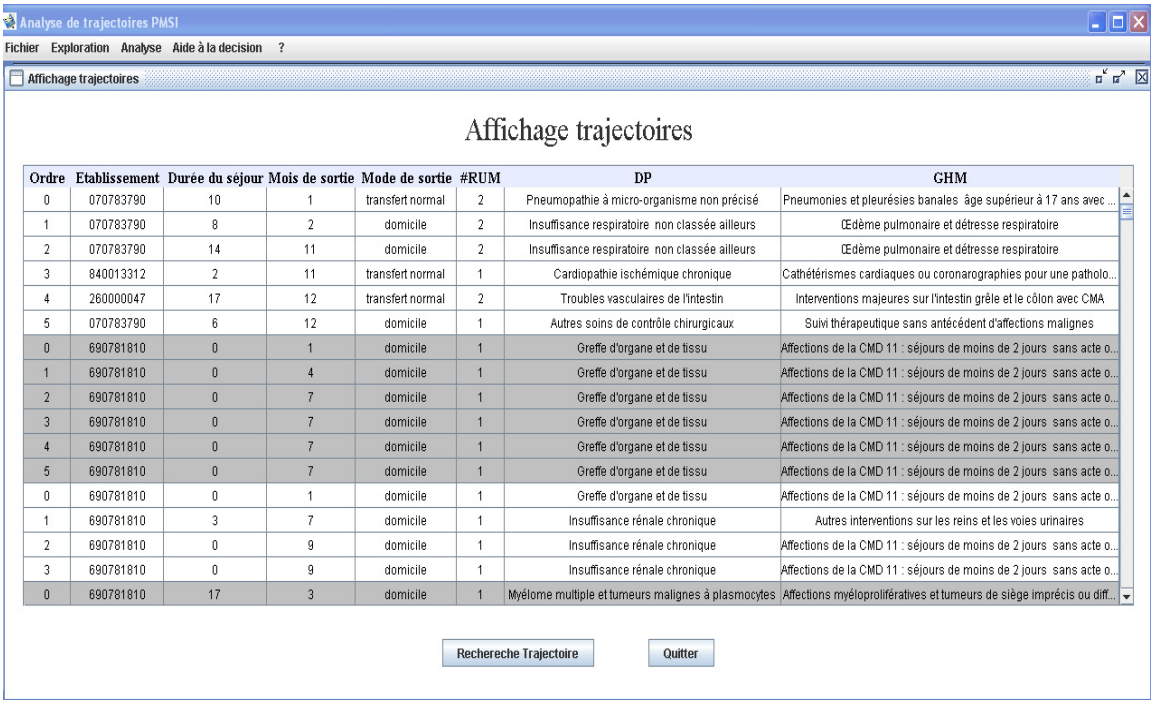



**Figure 5.2 - Affichage des trajectoires**

### 5.4.2. Le module de préparation et d'analyse

Ce module concerne l'implémentation des différents traitements de notre cadre d'analyse de données séquentielles du PMSI. Il assure ainsi l'exécution des opérations suivantes :

**La préparation des données**

Les données du PMSI, telles qu'elles sont fournies par l'ARH-RA (données brutes), ne permettent pas de lancer facilement les processus d'ajout, de suppression et d'analyse. En effet, ces données, provenant de différentes tables, sont parfois hétérogènes (de formats différents), complexes (classiques et symboliques) et leurs valeurs peuvent atteindre la taille d'un paragraphe dans le cas de données textuelles (à savoir, les libellés d'actes, des DPs et des GHMs). Pour exploiter de telles données à des fins décisionnelles, nous avons

généré des tables de données intermédiaires qui servent à l'analyse et notamment à la classification. Cette génération, faite d'une manière automatique, effectue une sorte de correspondance entre les tables des données (table de trajectoires, table de séjours, tables d'actes médicaux) pour faciliter l'ajout et la suppression des séjours et des trajectoires. Elle permet également de fournir un fichier comprenant la liste des triplets (id_traj : identifiant d'une trajectoire, DP, GHM) qui sera pris en entrée par l'algorithme de classification automatique par *b-coloration*.

**La classification automatique des trajectoires**

Dès que l'administrateur lance le module de classification, les processus de construction de la matrice des dissimilarités entre les trajectoires et de classification automatique par la *b-coloration de graphes* sont déclenchés. Cette opération prend en entrée le fichier retourné par le module de préparation de données, et fournit en sortie une typologie de trajectoires hospitalières dont les classes sont identifiées chacune par un ensemble de *trajectoires prototypes* dites aussi *profils de patients*.

**La modélisation par les chaînes de Markov**

Ce module assure l'exécution du processus de couplage entre les résultats de l'approche de classification par la b-coloration de graphes et le modèle de mélange par les chaines de Markov. Une fois ce processus accompli, les classes de trajectoires obtenues sont modélisées chacune par une chaine de Markov d'ordre 1. Les paramètres des modèles de classes étant nombreux, il est nécessaire de les stocker pour les exploiter dans les étapes ultérieures à des fins décisionnelles. Le langage XML (*eXtensible Markup Language*) s'avère une solution appropriée à cette étape primordiale. Les données sont alors stockées dans un fichier XML formé conformément à une grammaire associée, exprimée sous forme de DTD (*Document Type Definition*).

Le fichier XML est bien structuré de manière à contenir toutes les informations nécessaires pour le classement d'une nouvelle trajectoire ainsi que pour la prévision de la suite des séjours de soins futurs d'un patient. Le choix du langage XML peut être justifié par les points suivants :

- En cas d'utilisation de bases de données au lieu d'un fichier XML, l'interaction (avec la base) serait coûteuse (mise à jour et consultation), alors qu'avec un fichier XML, ces problèmes sont évités : l'interaction est plus rapide et l'espace de stockage est très faible.
- Si les données sont stockées dans des matrices statiques, ces dernières seront creuses (beaucoup de valeurs nulles). De plus, à chaque étape de prévision, tous les modules précédents seraient relancés pour pouvoir sélectionner les données nécessaires (absence de stockage physique pertinent), ce qui serait très couteux au niveau du temps d'exécution.

- Le fichier XML est utilisé dans l'étape d'évaluation des performances de la méthodologie. Ainsi, pour comparer deux approches différentes d'analyses de données séquentielles, il suffit de changer le fichier associé à l'une par l'autre, puis analyser les résultats obtenus.
- Le langage XML est un standard d'échanges. Nous pouvons utiliser notre fichier XML sur tout type d'environnement et tout type de machine (contrairement aux bases de données qui nécessitent des configurations et des licences spécifiques).

L'adoption de la solution XML dans notre plateforme semble dès lors totalement satisfaisante pour notre problématique.

### 5.4.3. Le module d'aide à la décision

Le module d'aide à la décision comprend les cinq composantes principales suivantes :

**La recherche**

La recherche d'une trajectoire est une étape primordiale avant l'étape de prévision. En effet, l'utilisateur doit préciser sur quel patient il désire faire la prévision. Une liste déroulante contenant tous les patients selon leurs numéros anonymes filtre la sous-liste des possibles dès la saisie d'une lettre ou d'un chiffre, dans le but de faciliter la tâche de recherche. L'utilisateur peut ainsi choisir le numéro anonyme du patient désiré et afficher les détails de sa trajectoire de soins sous la forme d'un tableau (voir figure 5.3). Une fois le patient choisi, l'opération de prévision de son futur épisode de soins peut être lancée.

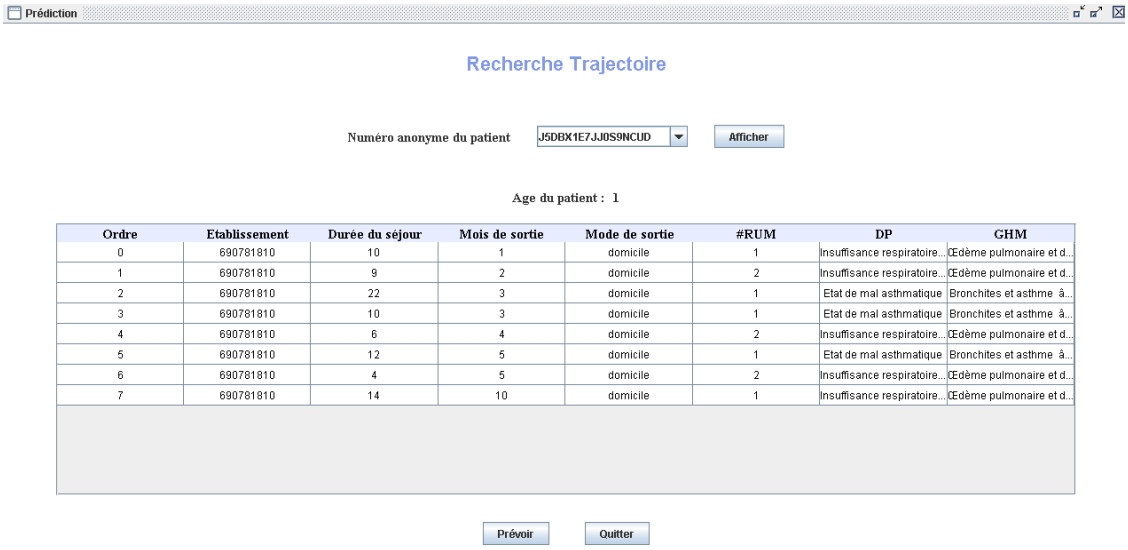



**Figure 5.3 - Module de recherche d'une trajectoire de soins**

**La prévision**

L'opération de prévision est l'une des étapes les plus importantes pour l'utilisateur de cette application. Une fois le patient choisi, le système peut prévoir le couple {groupe

homogène de malades, diagnostic principal} les plus probables pour le futur séjour. A ces informations est associée la probabilité estimée de passage de l'état courant vers ce nouvel état comme le montre la figure 5.4.

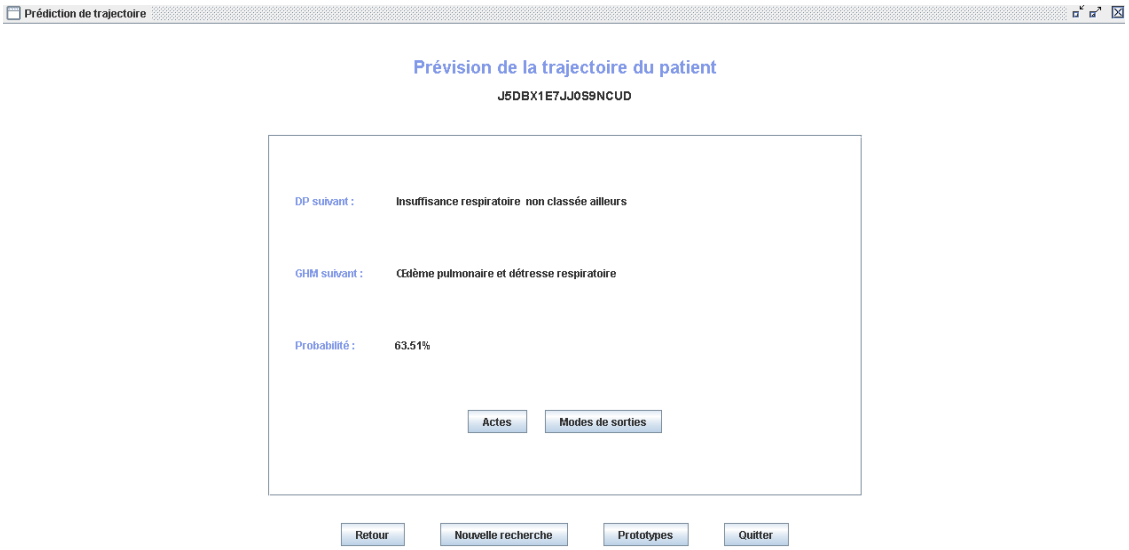



**Figure 5.4 - Prévision de trajectoire d'un patient**

L'opération de prévision ne se limite pas au groupe de malade et au diagnostic principal, mais peut concerner aussi les actes médicaux que peut subir le patient (*c.f.* figures 5.5) et le mode de sortie probable pour le prochain séjour (*c.f.* figure 5.6). Ce module propose également un affichage des séquences prototypes de la classe de la trajectoire en cours de prévision.

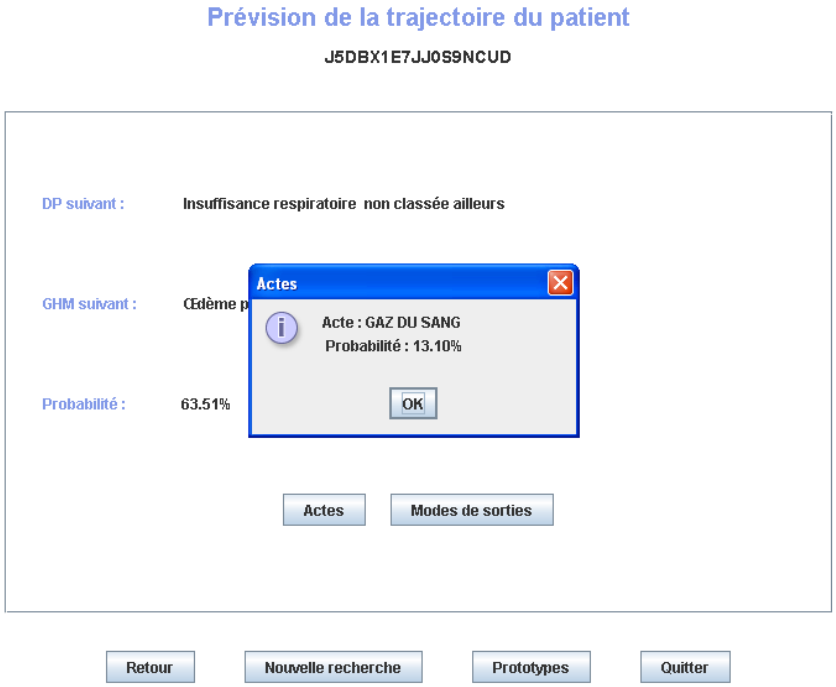



**Figure 5.5 – Estimation des actes médicaux à subir**

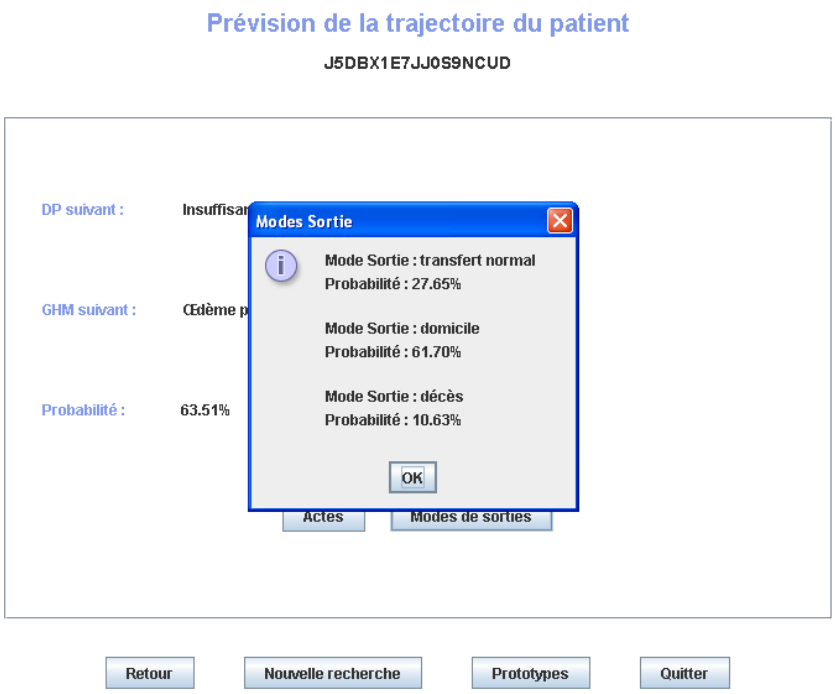



**Figure 5.6 - Mode de sortie probable**

### Evaluation du modèle pour la prévision

La plateforme *"Analyse de trajectoires PMSI"* propose également une composante qui permet l'analyse des performances du module de prévision sur la base de test ainsi que sur la base d'apprentissage (section 5.3.2). Les résultats sont obtenus via un graphique sous la forme de secteurs 3D, en fonction du nombre de bonnes/mauvaises prévisions, comme le montre la figure 5.7 suivante.

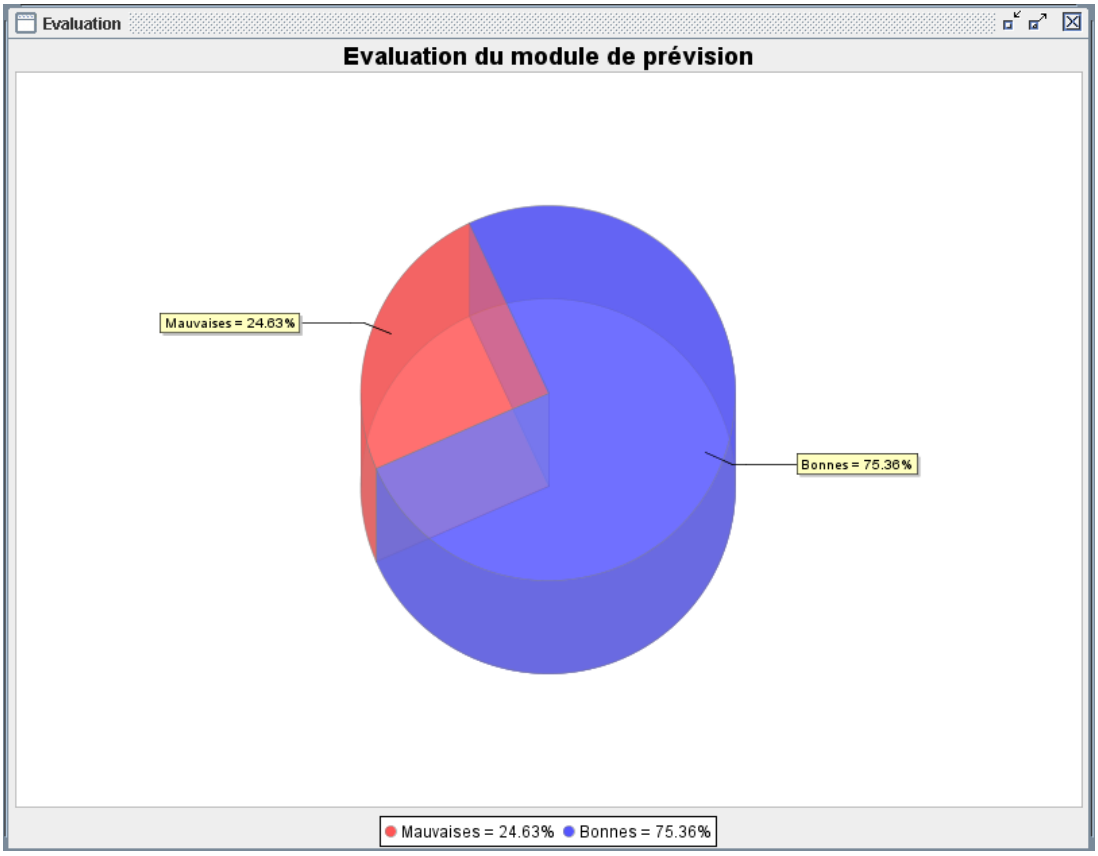



**Figure 5.7 - Evaluation de la prévision**

### Statistiques par classe de trajectoires

La plateforme permet également d'effectuer des statistiques sur les classes obtenues afin de mieux étudier les spécifications de chaque classe. Les statistiques concernent les variables Mode de sortie, Actes médicaux, Diagnostic principal (DP), Groupe homogène de malade (GHM), sexe et âge des patients appartenant à chaque classe (voir figure 5.8 pour les statistiques de la variable DP).

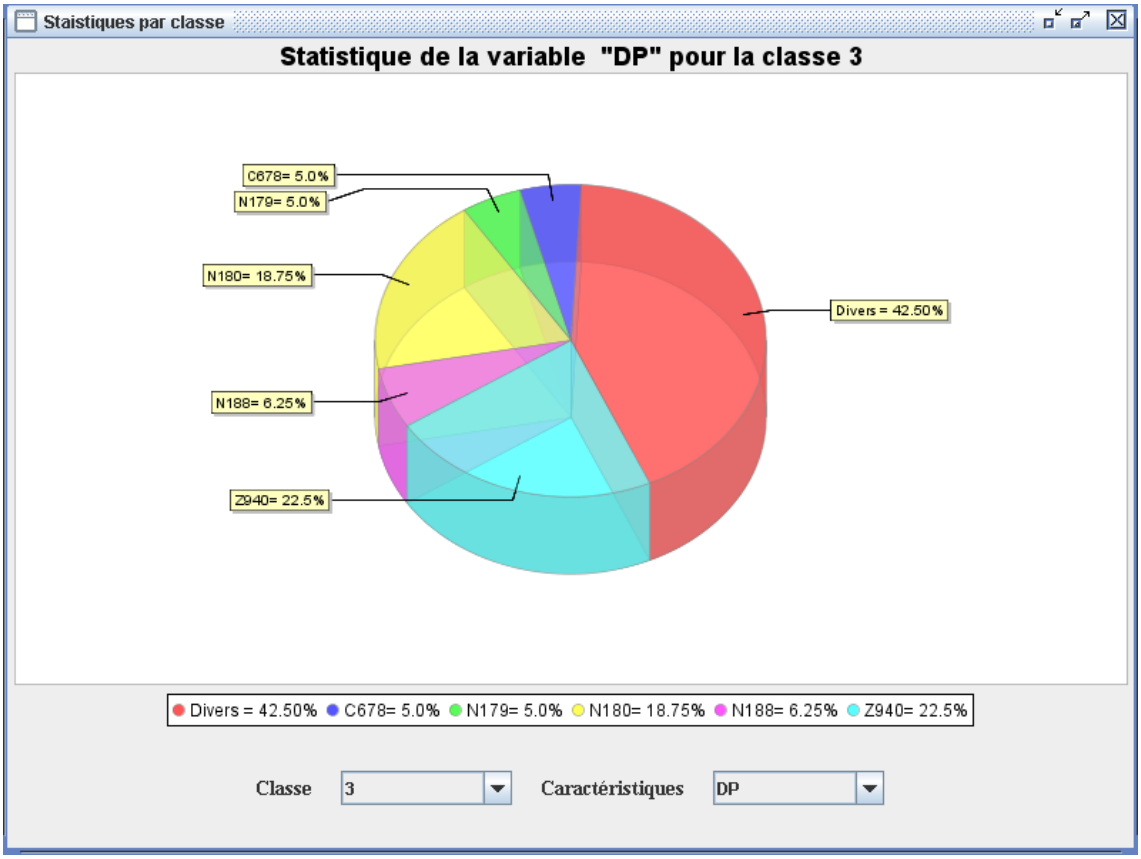
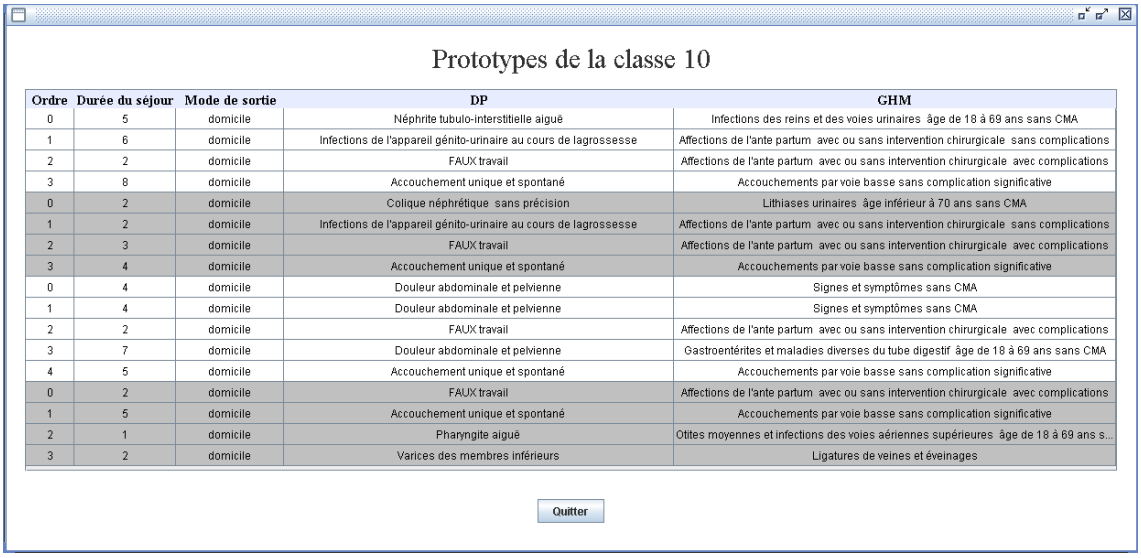



**Figure 5.8 - Statistique par classe**

Les séquences prototypes de chaque classe peuvent être à leur tour affichées. Elles caractérisent chaque classe et permettent d'offrir une idée sur les propriétés de cette classe et les profils de patients pouvant y appartenir (voir figure 5.9).



Prototypes de la classe 10

| Ordre | Durée du séjour | Mode de sortie | DP | GHM |
|---|---|---|---|---|
| 0 | 5 | domicile | Néphrite tubulo-interstitielle aiguë | Infections des reins et des voies urinaires âge de 18 à 69 ans sans CMA |
| 1 | 6 | domicile | Infections de l'appareil génito-urinaire au cours de lagrossesse | Affections de l'ante partum avec ou sans intervention chirurgicale sans complications |
| 2 | 2 | domicile | FAUX travail | Affections de l'ante partum avec ou sans intervention chirurgicale avec complications |
| 3 | 8 | domicile | Accouchement unique et spontané | Accouchements par voie basse sans complication significative |
| 0 | 2 | domicile | Colique néphrétique sans précision | Lithiases urinaires âge inférieur à 70 ans sans CMA |
| 1 | 2 | domicile | Infections de l'appareil génito-urinaire au cours de lagrossesse | Affections de l'ante partum avec ou sans intervention chirurgicale sans complications |
| 2 | 3 | domicile | FAUX travail | Affections de l'ante partum avec ou sans intervention chirurgicale avec complications |
| 3 | 4 | domicile | Accouchement unique et spontané | Accouchements par voie basse sans complication significative |
| 0 | 4 | domicile | Douleur abdominale et pelvienne | Signes et symptômes sans CMA |
| 1 | 4 | domicile | Douleur abdominale et pelvienne | Signes et symptômes sans CMA |
| 2 | 2 | domicile | FAUX travail | Affections de l'ante partum avec ou sans intervention chirurgicale avec complications |
| 3 | 7 | domicile | Douleur abdominale et pelvienne | Gastroentérites et maladies diverses du tube digestif âge de 18 à 69 ans sans CMA |
| 4 | 5 | domicile | Accouchement unique et spontané | Accouchements par voie basse sans complication significative |
| 0 | 2 | domicile | FAUX travail | Affections de l'ante partum avec ou sans intervention chirurgicale avec complications |
| 1 | 5 | domicile | Accouchement unique et spontané | Accouchements par voie basse sans complication significative |
| 2 | 1 | domicile | Pharyngite aiguë | Otites moyennes et infections des voies aériennes supérieures âge de 18 à 69 ans s... |
| 3 | 2 | domicile | Varices des membres inférieurs | Ligatures de veines et éveinages |

Quitter

**Figure 5.9 - Les séquences prototypes d'une classe de trajectoires**

### Statistiques sur la base totale de trajectoires

La plateforme propose aussi des statistiques sur la totalité de la base des trajectoires disponibles. Comme pour les classes des trajectoires, les statistiques concernent de la même façon l'âge, les actes, les modes de sorties, les DP, les GHM et le sexe des patients (voir figure 5.10) mais pour cette fois la totalité de la population donnée en entrée.

Ce type d'information est particulièrement utile pour nos partenaires médecins.

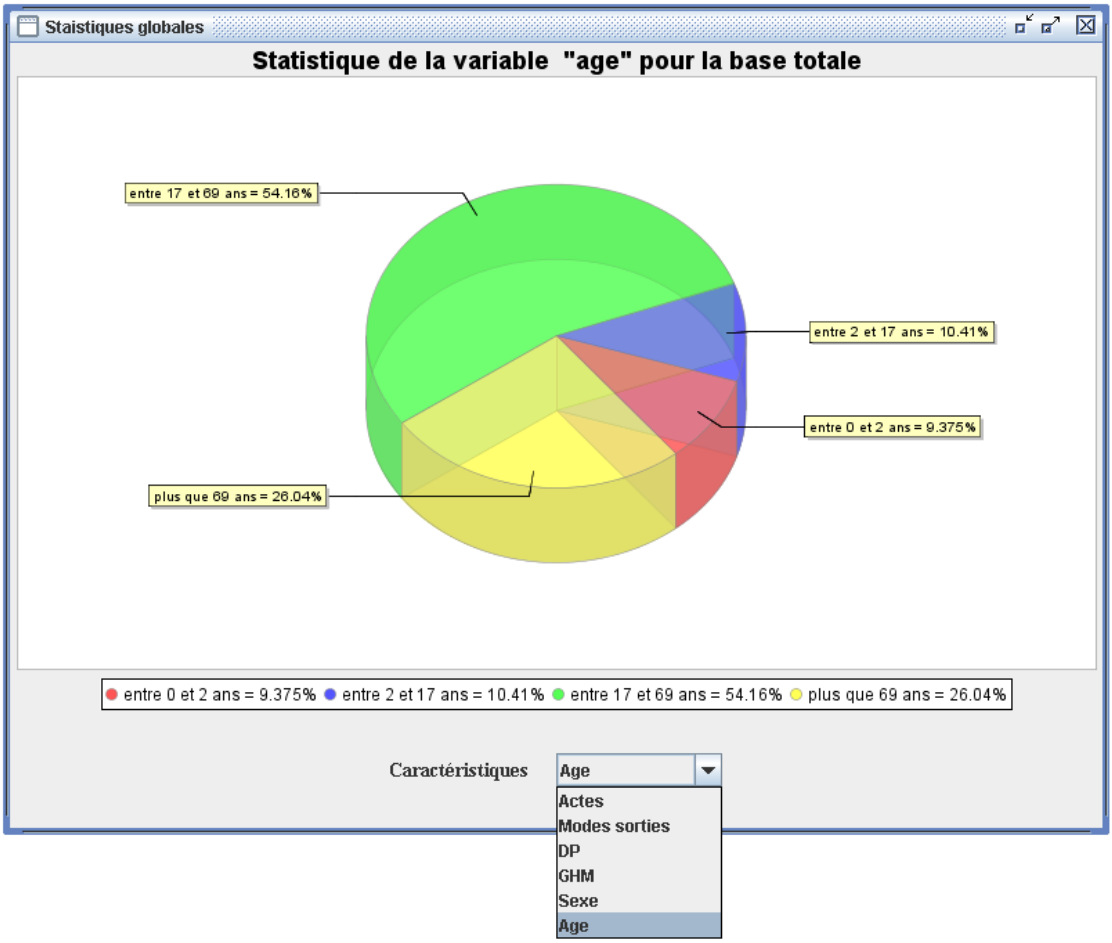



**Figure 5.10 - Statistique sur la base totale**

## 5.5. Conclusion

Dans ce chapitre, nous avons présenté un nouveau cadre d'analyse de données séquentielles qui permet la classification automatique et la prévision des séquences. Se basant sur une méthodologie mixte combinant l'approche de classification par *b-coloration de graphes* présentée dans le chapitre 3 et les chaines de Markov, il fournit une typologie des séquences en classes homogènes et bien séparées. Celles-ci sont représentées à la fois par un ensemble de *séquences-types* (*profils*) et un *modèle probabiliste de génération de données* (chaine de Markov) résumant les relations entre les états des séquences de la classe. Ce modèle assure notamment une meilleure *interprétabilité* des classes construites,

peut être appliqué pour classer de nouvelles séquences (*classement*) et estimer leurs évolutions futures (*prévision*).

Afin de valider notre démarche, nous avons proposé d'appliquer nos propositions sur des jeux de trajectoires extraits d'une base de données PMSI-2003 fournie par l'Agence Régionale d'Hospitalisation Rhône-Alpes (ARH-RA). Cette application nous a permis d'apprécier l'intérêt et la pertinence des résultats obtenus par rapport aux objectifs d'analyse. Un ensemble d'indices évaluant la qualité des résultats obtenus nous ont permis aussi d´évaluer les performances de notre approche comparées à une approche existante de la même famille (modèle de mélange markovien), sur le jeu réel de données médicales.

Enfin, nous avons présenté la plateforme logicielle *"Analyse de trajectoires PMSI"* que nous avons développée afin de concrétiser nos propositions théoriques dans le cadre du couplage entre l'approche de classification par la b-coloration de graphes présenté dans le chapitre 3 et le modèle de mélange de chaines de Markov. Il s'agit d'un outil d'aide à la décision dédié à la classification et à la prévision des trajectoires patient. L'objectif de cet outil est d'anticiper l'activité de l'établissement de santé pour lui permettre autant que possible de mieux organiser ses ressources (humaines et matérielles) et d'évaluer les coûts du séjour dès l'arrivée du patient. La plateforme nous est particulièrement utile aujourd'hui dans le processus de validation de la pertinence médicale des résultats obtenus, processus mené avec l'association des Directeurs de l'Information Médicale avec qui nous sommes en relation depuis quelques mois.

# Chapitre 6

# Conclusion générale

*" Je ne campe pas sur le passé, j'en tire des conclusions pour le présent. "*

Eric Fisher

## 6.1. Bilan des contributions

Dans le cadre de cette thèse, nous avons essayé d'apporter des solutions à la double problématique de *classification* de données complexes et *d'analyse de séquences temporelles*, solutions qui ont été appliquées aux données du système d'information hospitalier français.

L'objectif de la classification des données médicales était de fournir une typologie plus fine des séjours hospitaliers, ces séjours étant décrits selon un ensemble de données complexes, comme alternative à la classification en Groupes Homogènes de Malades (GHMs) actuelle. La nouvelle typologie devait répondre aux problèmes de non représentation des GHMs afin notamment de limiter les comportements opportunistes des établissements (point de vue des institutions nationales et régionales) et de permettre aux centres hospitaliers de disposer de budgets en concordance avec leur activité réelle (point de vue des établissements). La principale motivation fut ainsi de définir une méthode de classification automatique capable de traiter des données de descriptions complexes et hétérogènes (quand le nombre de classes n'est pas fixé à priori) qui permet en particulier de pouvoir facilement interpréter les classes obtenues à l'aide d'un ensemble représentatif d'individus.

Pour y parvenir, nous nous sommes basés sur une technique récente de la théorie des graphes baptisée *b-coloration*. Cette technique de coloration possède l'avantage de fournir une partition fine des données où la *séparation interclasse* est réalisée simultanément avec la *cohésion intraclasse*, et cela sans que le nombre de classes différentes représentant les individus ait été fixé à l'avance. La b-coloration possède également un ensemble de caractéristiques intéressantes, à savoir : (1) elle est applicable à tout type de données dès

lors qu'il est possible de construire une *matrice de dissimilarités* entre les individus à classer, et (2) pour chaque classe, elle identifie les *points dominants* qui sont le reflet des propriétés de la classe (ce qui donne du *sens* aux classes et en facilite ainsi l'interprétabilité), qui garantissent une *nette séparation* entre les classes, et qui assurent ainsi une grande stabilité à la partition obtenue en cas de modification (ajout/suppression) des individus de la population.

Une expérimentation a ainsi été présentée qui portait sur des jeux de données réelles et de données benchmark, les données réelles étant extraites de la base de données PMSI-2003 fournie par l'Agence Régionale d'Hospitalisation Rhône-Alpes (ARH-RA), et les résultats ont été comparés à ceux obtenus avec d'autres approches de classification. La nouvelle méthode de classification peut être utilisée pour différents types d'applications. A titre d'illustration, nous avons présenté le travail effectué dans le cadre d'un projet européen (**TArcHNA**[9]) sur une base d'images archéologiques. Ainsi pour les différentes expérimentations menées, les résultats obtenus sont significativement encourageants et ont montré l'intérêt de la méthode pour identifier des classes "homogènes" et "significatives".

Nous avons ensuite proposé une extension de l'approche de *classification* qui concerne l'apprentissage automatique, dans l'objectif d'affecter un nouvel individu (ou un groupe d'individus, *i.e.* de séjours patients) à la classe la plus adéquate, ou encore de retirer un individu, sans relancer la classification sur l'ensemble des données. Dans ce cadre, un nouvel algorithme de mise à jour incrémentale, se basant uniquement sur la connaissance des dissimilarités entre les individus pris deux à deux et sur la notion de dominance des classes, a été conçu. Un processus de validation exploitant différents cas d'ajouts de nouveaux individus a été présenté qui a montré la pertinence et l'efficacité de l'algorithme incrémental. Son utilisation pour les données médicales est possible mais le mode d'exploitation reste à être validé avec nos partenaires médicaux. L'idée serait de réévaluer les groupes de maladies obtenus à partir des nouvelles données envoyées par les établissements chaque semestre (ou chaque année). Ce processus permettrait d'ajuster la typologie en fonction de l'évolution des séjours, et de mieux affecter les budgets en cohérence avec l'évolution des traitements observée.

Dans la deuxième partie de cette thèse, nous nous sommes intéressés au problème d'analyse de séquences temporelles. Appliqué au domaine médical, il s'agissait de fournir une aide au pilotage stratégique des établissements de soins. Nous avons ainsi proposé un nouveau cadre d'analyse de données séquentielles qui permet la *classification automatique* et la *prévision* des séquences. Se basant sur une méthodologie mixte combinant notre approche de classification par *b-coloration de graphes* et les chaines de Markov, il fournit une typologie des séquences en classes homogènes et bien séparées. Celles-ci sont représentées à la fois par un ensemble de *séquences-types* (*profils*) et un *modèle*

[9] http://www.tarchna.org/home.htm

*probabiliste de génération de données* (chaine de Markov) résumant les relations entre les états des séquences de la classe. Ce modèle assure notamment une meilleure *interprétabilité* des classes construites et peut être appliqué pour classer de nouvelles séquences (*classement*), et estimer leurs évolutions futures (*prévision*).

Pour valider notre démarche, nous avons appliqué nos propositions sur des jeux de trajectoires patients (succession de séjours) extraits de la base de données PMSI-2003 et avons comparé les résultats avec ceux d'une autre approche de prévision de la même famille. Cette application nous a permis d'apprécier l'intérêt et la pertinence des résultats obtenus par rapport aux objectifs recherchés.

Sur un plan technique, nous avons développé une plateforme logicielle appelée *"Analyse de trajectoires PMSI"* en Java et XML, dans le but de proposer à nos partenaires (médecins ou responsables administratifs) un outil qui leur permette d'évaluer concrètement l'apport du cadre théorique proposé. Ce progiciel constitue donc un outil d'aide à la décision hospitalière dédié à la classification et à la prévision des trajectoires patients.

## 6.2. Perspectives de recherche

Les travaux réalisés dans cette thèse ouvrent diverses perspectives de recherche.

***Classification par b-coloration de graphes***

La méthodologie de classification non supervisée par b-coloration de graphes que nous avons proposée est itérative. Elle consiste, à chaque itération correspondante à un seuil de dissimilarité extrait de manière croissante de la matrice de dissimilarités entre les individus, à appliquer l'algorithme de *b-coloration* sur le *graphe seuil supérieur* en question, et à évaluer par la suite la qualité des partitions obtenues en utilisant un critère spécifique d'optimalité. L'objectif ici est d'identifier la meilleure partition qui sera renvoyée à l'utilisateur. Cette approche possède certaines limites, communes aux méthodes itératives de classification, dont le problème de la complexité lorsqu'on dispose de plusieurs valeurs de seuil dans la matrice des dissimilarités. Dans ce cadre, nous pensons proposer une heuristique permettant d'assouplir le processus itératif, en limitant le traitement sur certains seuils supposés pertinents, et sachant que deux seuils successifs peuvent parfois fournir la même partition.

D'autre part, pour améliorer la qualité des partitions obtenues, nous envisageons d'effectuer un réarrangement de certaines couleurs (classes). La stratégie étant de remettre en cause les classes de certains individus en modifiant leur couleur, à condition qu'une telle transformation ne viole pas les contraintes de la *b-coloration* et les bonnes qualités de partitionnement. Dans cette optique, un premier travail a été déjà effectué en collaboration avec M. Yoshida par la proposition d'un algorithme glouton de recoloration

des points dits *non critiques* (définition 3.2), dont la modification de couleurs ne va pas altérer les caractéristiques garanties par la *b-coloration* [Elghazel *et al.*, 2007e]. Des expérimentations ont été menées sur des jeux de données benchmark et les résultats obtenus on été comparés à ceux des approches classiques de classification. L'efficacité du nouvel algorithme glouton a ainsi été prouvée comme améliorant la qualité du partitionnement. Pour poursuivre encore cet objectif, une voie envisagée serait maintenant d'étendre le concept de recoloration aux *points critiques*, tout en garantissant les propriétés recherchées de la b-coloration.

Enfin, en vue d'analyser la souplesse de la méthode à d'autres objectifs de l'analyse de données, nous proposons de traiter un sujet en développement ces dernières années : *la classification sous-contraintes*. Dans ce cas, l'approche de classification cherche à produire une partition sur les données en exploitant des *connaissances a priori* dites *contraintes*. Une première réflexion dans cet axe a donné lieu à une modification de la méthode pour tenir compte de deux types de contraintes au niveau des individus : les contraintes *must-link* (deux individus doivent être dans la même classe) et les contraintes *cannot-link* (deux individus ne doivent pas être placés dans la même classe). Les premiers résultats obtenus sur des jeux de données benchmark sont très encourageants et ont montré l'intérêt des contraintes pour améliorer la précision du partitionnement [Elghazel *et al.*, 2007a].

### *Classification incrémentale*

Concernant l'approche *incrémentale* de classification automatique, des améliorations et des évolutions sont envisageables pour :

- mener d'autres tests et comparaisons, avec un ensemble d'approches de classification non supervisée, sur des jeux de données de plus grande taille, notamment ceux concernant les données médicales du PMSI,
- étendre le concept de *mise à jour incrémentale* en vue d'ajouter ou de supprimer simultanément un ensemble d'individus (permettant notamment la fusion de plusieurs partitions).

### *Analyse de données séquentielles*

Dans le cadre de nos travaux sur l'analyse de données séquentielles, nous allons chercher à améliorer la qualité de prévision de notre cadre d'analyse, en considérant cette fois des *chaines de Markov d'ordre k*, traduisant ainsi le fait que la probabilité de transition vers un état donné ne dépend plus uniquement de l'état précédent mais des *k* états qui le précèdent.

Une autre amélioration serait d'introduire un aspect sémantique entre les états des séquences, ce qui pourrait jouer un rôle important dans la classification et la modélisation des classes dans le but d'améliorer la qualité de la partition en groupes de séquences obtenue. L'idée étant d'enrichir notre méthodologie à partir de connaissances

supplémentaires spécifiques et notamment sur la *similarité* entre les différents états observés.

Enfin, dans la version actuelle de notre plateforme logicielle *"Analyse de trajectoires PMSI"*, nous nous sommes limités aux données d'une seule année du système PMSI (année 2003). Une prochaine étape pourrait être de travailler sur les données de trois années successives (2003, 2004 et 2005) afin d'exploiter plus largement le fichier de chaînage mis en place par l'ARH-RA, qui permet de retrouver la correspondance entre les numéros anonymes affectés aux patients.

# Bibliographie


Agrawal R. and Srikant R. *Mining Sequential Patterns*, Proceedings of the 11th International Conference on Data Engineering (ICDE'95), Tapei, Taiwan, 1995.

Ankerst M., Breunig M. M., Kriegel H. P. and Sander J. *OPTICS: Ordering points to identify clustering structure.*, Proceedings of the ACM SIGMOD Conference, Philadelphia, PA, pp. 49-60, 1999.

Antunes C. and Oliveira A. *Temporal data mining: an overview*, In KDD Workshop on Temporal Data Mining, pp. 1-13, 2001.

Auguston J. G. and Minker J. *An analysis of some graph theoretical clustering techniques*, Journal of ACM, **17**(4), pp. 571-588, 1970.

Azzaoui M., Legrand J. and Elghazel H. *KSyDoC: A Keyword-based System for DOcument Clustering and Retrieval*, Proceedings of the 23ème journées de Bases de données Avancées (BDA 2007), Démonstration, Marseille, France, 2007.

Baum L., Petrie T., Soules G. and Weiss N. *A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains*, The Annals of Mathematical Statistics, **41**, pp. 164-171, 1970.

Berkhin P. *Survey of clustering data mining techniques*, Accrue Software, 2002.

Bertrand P. and Diday E. *Une généralisation des arbres hiérarchiques : les représentations pyramidales*, Revue de Statistique Appliquée, **38**(3), pp. 53-78, 1990.

Bezdek J. C. and Pal N. R. *Some new indexes of cluster validity*, IEEE Transactions on Systems, Man and Cybernetics, **28**(3), pp. 301-315, 1998.

Bilme J. A. *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*, Technical ReportICSI-TR-97-021, University of Berkeley, 1998.

Bisson G. *La similarité: une notion symbolique/numérique*, In Apprentissage symbolique-numérique, éd. par Moulet B. Editions CEPADUES, pp. 169-201,2000.

Biswas G., Weinberg J. B. and Fisher D. H. *ITERATE : A conceptual clustering algorithm for data mining*, IEEE Transactions on Systems, Man and Cybernetics, **28**(C), pp. 219-230, 1998.

Blake C. L. and Merz C. J. *UCI repository of machine learning databases*, Available from http://www.ics.uci.edu/~mlearn/MLRepository.html (Octobre 2007), University of California, Irvine, Dept. of Information and Computer Sciences, 1998.

Bock H. H. *Dissimilarity measures for probability distributions*, In Analysis of symbolic data. Exploratory methods for extractin statistical information from complex data, éd. par Bock H. H. and Diday E. 153-165,2001.

Bock H. H. and Diday E. *Analysis of symbolic data. Exploratory methods for extracting statistical information from complex data,* Springer, 2001.

Box G. E. P. and Jenkins G. M. *Time Series Analysis, Forecasting and Control,* Holden-Day, San Francisco, 1970.

Box G. E. P., Jenkins G. M. and Reinsel G. C. *Time series analysis: Forecasting and control,* Prentice Hall, Englewood Cliffs, 1994

Buzan D., Sclaroff S. and Kollios G. *Extraction and clustering of motion trajectories in video*, Proceedings of the 17th International Conference on Pattern Recognition*,* **2**, pp. 521-524, 2004.

Cadez I. V., Gaffney S. and Smyth P. *A general probabilistic framework for clustering individuals and objects*, Proceedings of the Knowledge Discovery and Data Mining, pp. 140-149, 2000a.

Cadez I. V., Heckerman D., Meek C., Smyth P. and White S. *Visualization of navigation patterns on a Web site using model-based clustering*, Proceedings of the Knowledge Discovery and Data Mining, pp. 280-284, 2000b.

Celeux G., Diday E., Lechevallier Y., Govaert G. and Ralambondrainy H. *Classification automatique des données,* Editions Dunod, Paris, 1989.

Chatfield C. *The Analysis of Time Series,* Chapman & Hall, New York, 1996.

Chavant M. *A monothetic clustering method*, Pattern Recognition Letters*,* **19**(11), 1998.

Chavent M. *Analyse des données symboliques : une méthode divisive de classification*, PhD thesis, Thèse de doctorat, Université Paris-Dauphine, 1997.

Chavent M., Guinot C., Lechevallier Y. and Tenenhaus M. *Méthodes divisives de classification et segmentation non supervisée : recherche d'une typologie de la peau humaine saine*, Statistique Appliquée*,* **47**(4), pp. 87-99, 1999.

Cover T. and Hart P. *Nearest neighbor pattern classification*, IEEE Transactions on Information Theory*,* **13**(1), pp. 21-27, 1967.

De-Carvalho F. A. T. *Proximity coefficients between boolean symbolic objects*, In New Approaches in Classification and Data Analysis, éd. par Diday E., Lechevallier Y., Schader M., Bertrand P. and Burtschy B. Springer Verlag, pp. 387-394,1994.

Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K. and Harshman R. *Indexing by Latent Semantic Analysis*, Journal of the American Society for Information Science*,* **41**, pp. 391-407, 1990.

Dempster A. P., Laird N. M. and Rubin D. B. *Maximum likelihood from incomplete data via the EM-Algorithm*, Journal of the Royal Statistical Society, Series B*,* **39**, pp. 1-38, 1977.

Diday E. *La méthode des nuées dynamiques*, Statistiques Appliquées*,* **19**(2), pp. 19-34, 1971.

Diday E. *Une représentation visuelle des classes empiétantes : les pyramides*, Revue RAIRO APII*,* **20**(5), pp. 475-526, 1986.

Diday E. *Des objets de l'Analyse des Données à ceux de l'Analyse des Connaissances*, In Induction symbolique et numérique, éd. par Kodratoff Y. and Diday E. Cépadues, pp. 9-76,1991.

Diday E. and Govaert G. *Classification avec distances adaptatives*, Revue RAIRO*,* **11**(4), pp. 329-349, 1977.

Dubes R. C. *Cluster analysis and related issues*, In Handbook of pattern recognition and computer vision, éd. par Chen C. H., Pau L. F. and Wang P. S. P. World Scientific Publishing Co, pp. 3-32,1993.

Dunn J. C. *A fuzzy relative of the isodata process and its use in detecting compact, well-seperated clusters*, Journal of Cybernetics, **3**(3), pp. 32-57, 1973.

Durbin R., Eddy S., Krogh A. and Mitchison G. *Biological sequence analysis,* Cambridge University Press, Cambridge, UK, 1998.

Dussauchoy A. *Generalized information theory and decomposability of systems*, International Journal on General System, **9**, pp. 13-36, 1982.

Effantin B. and Kheddouci H. *The b-chromatic number of some power graphs*, Discrete Mathematics and Theoretical Computer Science, **6**, pp. 45-54, 2003.

Effantin B. and Kheddouci H. *A distributed algorithm for a b-coloring of a graph*, Proceedings of the International Symposium on Parallel and Distributed Processing and Applications (ISPA-06), Sorrento, Italy, pp. 430-438, 2006.

El-Golli A. *Extraction de données symboliques et cartes topologiques : Application aux données ayant une structure complexe*, PhD thesis, Thèse de doctorat, Université Paris-Dauphine, 2004.

El-Golli A., Conan-Guez B. and Rossi F. *Self-Organizing Map and symbolic data*, Journal of Symbolic Data Analysis, **2**(1), 2004.

Elghazel H. *Initiation au PMSI*, Rapport Interne, Laboratoire LIESP, 2005.

Elghazel H., Benabdeslem K. and Dussauchoy A. *Constrained Graph b-coloring based Clustering Approach*, Proceedings of the 9th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2007), Regensburg, Germany, Lecture Notes in Computer Science N°4654 - © Springer Verlag - ISBN: 978-3-540-74552-5, pp. 262-271, 2007a.

Elghazel H., Deslandres V. and Dussauchoy A. *Analyse de données PMSI : Etude des groupes homogènes de malades et proposition d'une nouvelle typologie de séjours*, Proceedings of the 3ème conférence francophone en gestion et ingénierie des systèmes hospitaliers (GISEH 2006), Luxembourg, 2006a.

Elghazel H., Deslandres V., KALLEL K. and Dussauchoy A. *Clinical Pathway Analysis Using Graph-Based Approach and Markov Models*, Proceedings of the Second IEEE International Conference on Digital Information Management (ICDIM07), Lyon, France, 2007b.

Elghazel H., Hacid M. S., Deslandres V., Dussauchoy A. and Kheddouci H. *A New Clustering Approach for Symbolic Data and its Validation: Application to the Healthcare Data*, Proceedings of the In 16th International Symposium on Methodologies for Intelligent Systems (ISMIS 2006), Bari, Italie, Lecture Notes in Computer Science N°4203 - © Springer Verlag - ISBN: 3-540-45764-X, pp. 473-482, 2006b.

Elghazel H., Hacid M. S., Kheddouci H. and Dussauchoy A. *A New Clustering Approach for Symbolic Data: Algorithms and Application to Healthcare Data*, Proceedings of the 22ème journées de Bases de données Avancées (BDA 2006), Lille, France, 2006c.

Elghazel H., Kheddouci H., Deslandres V. and Dussauchoy A. *A Partially Dynamic Clustering Algorithm for Data Insertion and Removal*, Proceedings of the 10th International Conference on Discovery Science (DS 2007), Sendai, Japan, Lecture Notes in Computer Science N° 4755- © Springer Verlag- ISBN 978-3-540-75487-9, pp. 78-90, 2007c.

Elghazel H., Kheddouci H., Deslandres V. and Dussauchoy A. *Une approche incrémentale de classification non supervisée par b-coloration de graphes*, Proceedings of the 9ème

Conférence Francophone sur l'apprentissage automatique(CAp 2007), Plate-forme AFIA, Editions Cépaduès, Grenoble, France, 2007d.

Elghazel H., Yoshida T., Deslandres V., Hacid M. S. and Dussauchoy A. *A New Greedy Algorithm for improving b-Coloring Clustering*, Proceedings of the 6th IAPR-TC-15 Workshop on Graph-based Representations in Pattern Recognition (GbR 2007), Alicante, Spain, Lecture Notes in Computer Science N°4538 - © Springer Verlag - ISBN: 978-3-540-72902-0, pp. 228-239, 2007e.

Esposito F., Malerba D. and Tamma V. *Dissimilarity measures for symbolic objects*, In Analysis of symbolic data. Exploratory methods for extractin statistical information from complex data, éd. par Bock H. H. and Diday E. 165-185,2001.

Estacio-Moreno A., Artières T. and Gallinari P. *Classification automatique de données biographiques*, Proceedings of the RJCIA 2005, Nice, France, 2005.

Estacio-Moreno A., Barbary O., Gallinari P. and Piron M. *Classification de données biographiques : application à des trajectoires migratoires vers Cali (Colombie)*, Revue de Statistique Appliquée, **52**(4), pp. 33-54, 2004.

Ester M., Kriegel H. P., Sander J. and Xu X. *A density-based algorithm for discovering clusters in large spatial databases with noise.* , Proceedings of the 2nd ACM SIGKDD, Portland, Oregon, pp. 226-231, 1996.

Fisher D. *Knowledge Acquisition via Incremental Conceptual Clustering*, Machine Learning, **2**, pp. 139-172, 1987.

Forgy E. W. *Cluster Analysis of Multivariate Data : Efficiency Versus Interpretability of Classifications*, Biometrics, **21**, pp. 768-780, 1965.

Forney G. D. *The viterbi algorithm*, **61**(3), pp. 268-278 1973

Fraley C. and Raftery A. E. *How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis*, Computer Journal, **41**, pp. 578-588, 1998.

Gomory R. E. and Hu T. C. *Multy-terminal Network flows*, Journal of SIAM, **9**(4), pp. 551-570, 1961.

Gotlieb C. C. and Kumar S. *Semantic clustering of index terms*, Journal of ACM, **15**(4), pp. 493-513, 1968.

Gowda K. C. and Diday E. *Symbolic clustering using a new dissimilarity measure*, Pattern Recognition, **24**(6), pp. 567-578, 1991.

Guénoche A., Hansen P. and Jaumard B. *Efficient algorithms for divisive hierarchical clustering with the diameter criterion*, **8**, pp. 5-30, 1991.

Guha S., Rastogi R. and Shim K. *CURE: An efficient clustering algorithm for large databases*, Proceedings of the ACM SIGMOD Conference, Seattle, WA, pp. 73-84, 1998.

Guha S., Rastogi R. and Shim K. *ROCK: A robust clustering algorithm for categorical attributes*, Information Systems, **25**(5), pp. 345-366, 2000.

Han J., Kamber M. and Tung A. K. H. *Spatial clustering methods in data mining*, In Geographic Data Mining and Knowledge Discovery, éd. par Miller H. and Han J. Taylor and Francis, pp. 1-29.,2001.

Hansen P. and Delattre M. *Complete-link cluster Analysis by graph coloring*, Journal of the American Statistical Association, **73**, pp. 397-403, 1978.

Hartigan J. and Wong M. *Algorithm AS136: A k-means clustering algorithm*, Journal of Applied Statistics, **28**, pp. 100-108, 1979.

Hartuv E. and Shamir R. *A clustering algorithm based on graph connectivity*, Information Processing Letters, **76**, pp. 175-181, 2000.

Hastie T., Tibshirani R. and Friedman J. *The elements of statistical learning: Data mining, inference and prediction,* Springer-Verlag, New York, 2001.

Ichino M. and Yaguchi H. *Generalized Minkowsky metrics for mixed feature type data analysis*, IEEE Transactions on Systems, Man and Cybernetics, **24**(4), pp. 698-708, 1994.

Irving W. and Manlove D. F. *The b-chromatic number of a graph*, Discrete Applied Mathematics, **91**, pp. 127-141, 1999.

Jaccard P. *Nouvelle recherche sur la distribution florale*, Bulletin Society Vaud, Science Natural, **44**, pp. 223-270, 1908.

Jain A. k., Murty M. N. and Flynn: P. J. *Data Clustering: A Review*, ACM Computing Surveys, **31**, pp. 264-323, 1999.

Jambu M. *Introducion au Data Mining,* Editions Eyrolles, Paris, 1999.

Juang B.-H. and Rabiner L. *A probabilistic distance measure for hidden markov models*, AT&T Technical Journal, **64**(2), pp. 391-408, 1985.

Karypis G., Han E. and Kumar V. *Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling*, IEEE Computer, **32**(8), pp. 68-75, 1999.

Kaufman L. and Rousseeuw P. J. *Finding groups in data : An introduction to cluster analysis,* John Wiley and Sons, New York, 1990.

Kheddouci H. *Placement et Paramètres de Graphes*, Mémoire HDR, Université de Bourgogne, 2003.

King J. R. *Machine-component grouping in production flow analysis: an approach using rank order clustering algorithm*, International Journal of Production Research, **18**(2), pp. 213-232, 1980.

Kodratoff Y., Napoli A. and Zighed D. A. *Bulletin AFIA, ECBD,* 2001.

Kohonen T. *Self-organising maps,* New York, 1997.

Koskela T., Lehtokangas M., Saarinen J. and Kaski K. *Time series prediction with multilayer perceptron, FIR and Elman neural networks*, Proceedings of the World Congress on Neural Networks, pp. 491-496, 1996.

Kouider M. and Maheo M. *Some bounds for the b-chromatic number of a graph*, Discrete Mathematics, **256**, pp. 267-277, 2002.

Kruskall J. B. and Liberman M. *The symmetric time warping problem: From continuous to discrete*, In Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison, éd. par Kruskal J. B. and Sankoff D. Stanford: CSLI Publications, pp. 125-161,1999.

Kuhns J. L. *Mathematical analysis of correlation clusters*, In Word correlation and automatic indexing, Progress Rep, éd. par Wooldridge R. and Park C., California,1959.

Laxman S. and Sastry P. S. *A Survey of Temporal Data Mining*, Sadhana, **31**(2), pp. 173-198, 2006.

Limam M. *Méthode de description de classes combinant classification et discrimination*, PhD thesis, Thèse de doctorat, Université Paris-Dauphine, 2005.

Mac-Queen J. B. *Some methods for classification and analysis of multivariate observations*, Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, **1**, Berkeley, University of California Press, pp. 281-297, 1967.

Malerba D., Esposito F., Gioviale V. and Tamma V. *Comparing Dissimilarity Measures for Symbolic Data Analysis*, Proceedings of the Joint Conferences on New Techniques and Technologies for Statistics and Exchange of Technology and Know-how, pp. 473-481, 2001.

Mali K. and Mitra S. *Clustering and its validation in a symbolic framework*, Pattern Recognition Letters, **24**(14), pp. 2367-2376, 2003.

Matula D. W. *Cluster Analysis via Graph Theoretic Techniques*, Proceedings of the Louisiana Conference on Combinatorics, Graph Theory, and Computing, University of Manitoba, Winnipeg, pp. 199-212, 1970.

Matula D. W. *K-Components, Clusters, and Slicings in Graphs*, SIAM Journal on Applied Mathematics, **22**, pp. 459-480, 1972.

Michalski R. S. and Stepp R. E. *Learning from observations : Conceptual clustering*, In Machine Learning: An Artificial intelligence approach, éd. par Michalski R. S., Carbonell J. G. and Mitchell T. M. Morgan Kaufmann, pp. 331-363,1983.

Nakache J. P. and Confais J. *Approche pragmatique de la classifciation,* Editions Technip, Paris, 2005.

NG R. and Han J. *Efficient and effective clustering methods for spatial data mining*, Proceedings of the 20th Conference on VLDB, Santiago, Chile., pp. 144-155, 1994.

NG R. and Han J. *CLARANS : A method for clustering objects for spatial data mining*, IEEE Transactions on Knowledge and Data Engineering, **14**(5), pp. 1003-1016, 2002.

Oates T., Firoiu L. and Cohen P. *Clustering time series with hidden Markov models and dynamic time warping*, Proceedings of the IJCAI-99 Workshop on Neural, Symbolic and Reinforcement Learning Methods for Sequence Learning, pp. 17-21, 1999.

Paterson M. and Dancik V. *Longest Common Subsequences*, Mathematical Foundations of Computer Science, **841**, pp. 127-142, 1994.

Quantin C., Mathy C., Brunet-Lecomte P., Metral P., Bismuth M. J., Dusserre L. and Gadreau M. *Proposition d'une modélisation de l'hétérogénéité de l'activation médicale pour améliorer la gestion hospitalière par groupes homogènes de malades*, Available from http://ungaro.u-bourgogne.fr/quantin/e9804.pdf (Octobre 2007), 1999.

Rabiner L. *A tutorial on hidden markov models and selected applications in speech recognition*, Proceedings of the IEEE, **77**(2), pp. 257-286, 1989.

Rand W. M. *Objective criteria for the evaluation of clustering methods*, Journal of the American Statistical Association., **66**, pp. 846-850, 1971.

Rote G. *Computing the minimum Hausdorff distance between two point sets on a line under translation*, Information Processing Letters, **38**, pp. 123-127, 1991.

Rousseeuw P. J. *Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis*, Journal of Computational and Applied Mathematics, **20**, pp. 53-65, 1987.

Sakoe H. *Two-level DP matching - a dynamic programming based pattern matching algorithm for connected word recognition*, IEEE Transactions on Acoustics, Speech, and Signal Processing, **27**(6), pp. 588-595, 1979.

Saunier N. and Sayed T. *Clustering Vehicle Trajectories with Hidden Markov Models Application to Automated Traffic Safety Analysis*, Proceedings of the International Joint Conference on Neural Networks ( IJCNN '06), pp. 4132- 4138, 2006.

Schroeder A. *Analyse d'un mélange de distributions de probabilité de même type.* , Revue de Statistique Appliquée, **24**(1), pp. 39-62, 1976.

Shi J. and Malik J. *Normalized cuts and image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **22**(8), pp. 888-905., 2000.

Sokal R. R. and Michener C. D. *Statistical method for evaluating systematic relationships*, University of Kansas science bulletin, **38**, pp. 1409-1438, 1958.

Sokal R. R. and Sneath P. H. A. *Principles of numerical taxonomy,* Freeman, San Francisco, 1963.

Touati M., Rahal M., Quantin C., Leteuff G., Diday E., Afonso F., Battaglia G. and Limam M. *Analyse de trajectoires hospitalières de patients atteints d'un infarctus aigu du myocarde*, Proceedings of the 6ème conférence francophone Extraction et Gestion des Connaissances (EGC 2006), Lille, France, 2006.

Tufféry S. *Data Mining et statistique décisionnelle L'intelligence dans les bases de données,* Editions TECHNIP, Paris, 2005.

Wu C. F. J. *Annals of statistics*, **11**(1), pp. 95-103, 1983.

Wu Z. and Leahy R. *An optimal graph theoretic approach to data clustering : Theory and its application to image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **15**(11), pp. 1101-1113, 1993.

XU X., ESTER M., KRIEGEL H. P. and SANDER J. *A distribution-based clustering algorithm for mining in large spatial databases*, Proceedings of the 14th ICDE, Orlando, FL, pp. 324-331 1998.

Zahn C. T. *Graph-theoretical methods for detecting and describing gestalt clusters*, IEEE Transactions on Computers, **20**(1), pp. 68-86, 1971.

Zhang T., Ramakrishnan R. and Livny M. *BIRCH: an efficient data clustering method for very large databases*, ACM SIGMOD Record, **25**(2), pp. 103-114, 1996.

# Résumé

D'une manière générale, la fouille de données regroupe l'ensemble des techniques soit descriptives (qui visent à mettre en évidence des informations présentes mais cachées par le volume des données), soit prédictives (cherchant à extrapoler de nouvelles connaissances à partir des informations présentes dans les données). Dans le cadre de cette thèse, nous nous intéressons au problème de classification et de prévision de données hétérogènes, que nous proposons d'étudier à travers deux approches principales. Dans la première, il s'agit de mettre en place une nouvelle approche de classification automatique basée sur une technique de la théorie des graphes baptisée b-coloration. Nous avons également développé l'apprentissage incrémental associé à cette approche, ce qui permet à de nouvelles données d'être automatiquement intégrées dans la partition initialement générée sans avoir à relancer la classification globale. Le deuxième apport de notre travail concerne l'analyse de données séquentielles. Nous proposons de combiner la méthode de classification précédente avec les modèles de mélange markovien, afin d'obtenir une partition de séquences temporelles en groupes homogènes et significatifs. Le modèle obtenu assure une bonne interprétabilité des classes construites et permet d'autre part d'estimer l'évolution des séquences d'une classe donnée.

Ces deux propositions ont ensuite été appliquées aux données issues du système d'information hospitalier français (PMSI), dans l'esprit d'une aide au pilotage stratégique des établissements de soins. Ce travail consiste à proposer dans un premier temps une typologie plus fine des séjours hospitaliers pour remédier aux problèmes associés à la classification existante en groupes homogènes de malades (GHM). Dans un deuxième temps, nous avons cherché à définir une typologie des trajectoires patient (succession de séjours hospitaliers d'un même patient) afin de prévoir de manière statistiques les caractéristiques du prochain séjour d'un patient arrivant dans un établissement de soins. La méthodologie globale offre ainsi un environnement d'aide à la décision pour le suivi et la maîtrise de l'organisation du système des soins.



**Mots clés :** Classification automatique, prévision, classification incrémentale, données hétérogènes, données séquentielles, b-coloration de graphes, séjours hospitaliers, trajectoires hospitalières.



# Abstract

Data mining generally includes either descriptive techniques (which provide an attractive mechanism to automatically find the hidden structure of large data sets), or predictive techniques (able to unearth hidden knowledge from datasets). In this work, the problem of clustering and prediction of heterogeneous data is tackled by a two-stage proposal. The first one concerns a new clustering approach which is based on a graph coloring method, named b-coloring. An extension of this approach which concerns incremental clustering has been added at the same time. It consists in updating clusters as new data are added to the dataset without having to perform complete re-clustering. The second proposal concerns sequential data analysis and provides a new framework for clustering sequential data based on a hybrid model that uses the previous clustering approach and the Mixture Markov chain models. This method allows building a partition of the sequential dataset into cohesive and easily interpretable clusters, as well as it is able to predict the evolution of sequences from one cluster.

Both proposals have then been applied to healthcare data given from the PMSI program (French hospital information system), in order to assist medical professionals in their decision process. In the first step, the b-coloring clustering algorithm has been investigated to provide a new typology of hospital stays as an alternative to the DRGs classification (Diagnosis Related Groups). In a second step, we defined a typology of clinical pathways and are then able to predict possible features of future paths when a new patient arrives at the clinical center. The overall framework provides a decision-aid system for assisting medical professionals in the planning and management of clinical process.



**Keywords :** Clustering, prediction, incremental clustering, heterogeneous data, sequential data, graph b-coloring, hospital stays, clinical pathways.