UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE APPLIQUÉES

PAR
ZINE-EL-ABIDINE SOUDANI

ÉTUDE COMPARATIVE DES ALGORITHMES DEDIES A LA CLASSIFICATION

DÉCEMBRE 2005

TABLE DES MATIÈRES

Liste des figures

Numéro **page**

Liste des tables

Numéro **page**

Remerciements

Ce travail a commencé au laboratoire du DMI de l'UQTR pour prendre sa vitesse de croisière et toute son ampleur au sein du laboratoire du CRED au CHU de Sherbrooke. Je tiens à remercier tout d'abord le bienfaiteur pour tout ce qu'il m'a permis de réaliser.

Mes intentions vont aux professeurs Ismail Biskri et Boucif Amar Bensaber, mes directeurs de recherche pour leur aide précieuse, leur patience et l'intérêt qu'ils portaient à mon travail.

Que l'équipe du CRED, à sa tête son directeur Andrew Grant, les chercheurs Faiza Boughrassa, Andriy Moshyk et Charaf Ahnadi, acceptent mes remerciements les plus sincères pour leur collaboration et leur dévouement désintéressé.

Je remercie, aussi, les professeurs François Meunier et Jean-François Quessy pour avoir accepté de lire et évaluer mon mémoire.

Je ne terminerai pas sans avoir une pensée pour tout le personnel et étudiants du département de mathématiques et informatique appliquées. Je pense notamment à tous les professeurs qui font un travail extraordinaire.

À ma mère Aïcha et mon père Fatah pour tout ce qu'ils m'ont donné

À ma femme Amel pour sa patience et son dévouement

À ma sœur Mounia et mes frères Mahmoud et Bensalem pour ce qu'ils représentent pour moi

À toute ma famille

À tous mes amis

Chapitre I

INTRODUCTION

1. LA CONNAISSANCE ET LE SAVOIR

La maîtrise de l'information est considérée comme l'une des plus importantes tâches de l'entreprise moderne, des gouvernements et des organismes de tous genres. Elle a pour but de mieux comprendre l'environnement pour pouvoir proposer, en conséquence, de meilleurs services. Elle sert aussi à prédire les évolutions futures afin de s'y adapter et être le plus efficace et le plus compétitif possible. L'information est représentée dans son état le plus élémentaire sous la forme de données. Une donnée informatique est une suite de signes alphanumériques et binaires interprétables et ayant un sens. Cette suite de signes est facilement utilisable par une machine informatique. La donnée n'est utile que pour supporter et représenter en association avec d'autres données, l'entité plus complexe qu'est l'information. Seul l'esprit humain est capable de distinguer, à partir d'un ensemble de données intelligibles, l'information sous-jacente, en allant vérifier la cohérence et le sens caché. L'information ne suffit pas à elle-même et ne vaut que si elle est comprise et interprétée de la bonne manière. L'information en soi n'a donc qu'un intérêt très relatif. Elle n'est que le support qui permet d'accéder à un concept plus général et plus utile pour l'être humain; c'est le concept de la connaissance. Un ensemble d'informations permet de générer, de reconstituer ou d'enrichir une connaissance sur un sujet.

Au début de notre introduction, nous aurions dû utiliser le terme «*maîtrise de la connaissance*» plutôt que «*maîtrise de l'information*». Notre erreur est délibérée et ne fait que se conformer à la réalité du monde, lequel, depuis l'avènement de l'outil informatique, s'est contenté de gérer des données, ou au mieux de l'information sans jamais penser à la connaissance qui est l'essence et le pourquoi de l'information. Avec l'incroyable accélération de la production de l'information et sa disponibilité presque sans limite, le monde commence à prendre conscience et s'intéresse de plus en plus à la connaissance que recèle toute cette information. Le concept de *base de connaissances* a été admis dans le jargon informatique pour compléter et généraliser le concept de la *base de données*, lequel reste une notion forte des deux dernières décennies. Des outils de questionnement des bases de données pour rechercher de l'information et certains outils statistiques permettant de résumer des aspects de ces mêmes bases de données, on a transité vers des outils plus subtils à comprendre et à manipuler. On est passé aux outils d'extraction de la connaissance à partir des bases de données. La question posée est de vérifier l'efficacité de tels outils et leurs capacités d'extraire les formes de connaissances. Encore faut-il recenser et définir toutes ces dernières. Une autre question plus futuriste est de penser à modéliser le *savoir*. Un ensemble organisé de connaissances sur un même sujet constitue un savoir. Nous pouvons alors dire qu'une gestion efficace des données, de l'information et des connaissances, devrait aboutir à la capacité de modéliser cette organisation de connaissances en un savoir identifiable. Cette modélisation sera supportée par des bases de connaissances. On serait même tenté de dire qu'on dépasserait le stade des bases de connaissances pour produire une nouvelle génération de *systèmes experts* que leur discipline-mère l'*intelligence artificielle* n'a pas su porter à la réussite lors de leur première apparition dans les années quatre-vingt.

La science de la représentation de la connaissance, dans les formes qu'elle prend aujourd'hui, n'est qu'à ses débuts. Les outils technologiques actuels sont loin de pouvoir modéliser l'intelligence humaine, source et moteur de la connaissance. Synthétiser l'information en connaissance et la connaissance en savoir reste à ce jour la spécialité jalousement gardée de l'être humain.

Avant d'aborder notre sujet, nous nous devons d'apporter quelques éclaircissements en donnant les définitions de certaines notions que nous allons manipuler tout au long de ce mémoire.

2. PROBLÉMATIQUE

Durant les 30 dernières années les performances des outils informatiques ont été multipliées par un facteur de plusieurs milliers. À titre d'exemple, le premier ordinateur du milieu des années 50 pesait 50 tonnes, consommait 25 kilowatts, avait quelques milliers de positions mémoire et exécutait une centaine d'instructions par seconde. Alors que le microprocesseur *Pentium D d'Intel*, avec ses 100 millions de transistors, pèse à peine quelques grammes, son boîtier mesure environ 24mm, ne consomme pas plus de 25 watts, gère jusqu'à 2 gigaoctets de mémoire vive, a une fréquence interne de 3,2 GHz (soit quelques 3,2 milliards d'instructions par seconde) et une fréquence externe de 800MHz [IntelPentiumD05]. Le *Pentium D*, bolide d'aujourd'hui apparaîtra aussi démodé que le premier ordinateur dans moins de 10 ans.

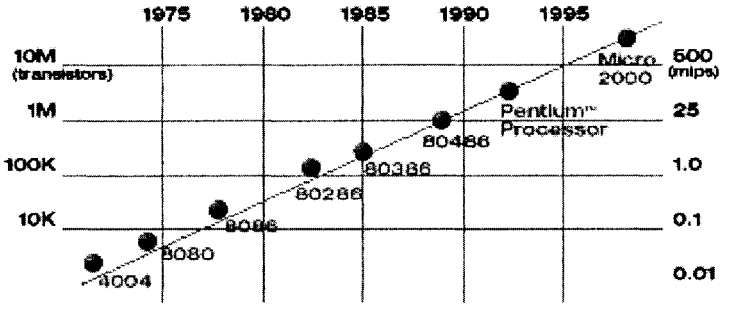

Figure 01. La performance des microprocesseurs double tous les 18 mois
***« source Intel au* http://www.intel.com»**

Cette incroyable évolution aussi matérielle que logicielle a contribué à générer de plus en plus d'information aussi variés qu'hétérogènes (Près de 3 million de pages s'ajoutent quotidiennement au Web). La vente de détail, les banques, les compagnies d'assurance ainsi que les hôpitaux sont parmi les secteurs qui ont généré le plus d'informations pour leurs besoins de fonctionnement. Les trois premiers cités, vu la grande concurrence qui règne dans leurs domaines respectifs, ont su utiliser les données qu'ils ont accumulées pour élaborer de meilleures stratégies de développement et une plus juste compréhension des besoins et attentes de leurs clients. Pour preuve, les règles d'association, technique de datamining, sont le fruit des recherches initiées par les vendeurs de détail.

Le monde médical n'a pu suivre cette évolution que très médiocrement. Les questions d'éthique expliquent mal ce retard. Le Centre Hospitalier Universitaire de Sherbrooke, fleuron des hôpitaux canadiens et pionnier dans la gestion électronique des dossiers patient, malgré le fait qu'il ait développé un entrepôt de données nommé CIRESSS[1] pour y archiver toutes les données historisées, n'échappe pas à la règle. L'entrepôt susnommé, malgré ses performances techniques et sa mise à jour quotidienne, reste sous-utilisé et l'information qu'il contient méconnue.

Notre travail consiste tout d'abord à défricher le terrain et à vérifier l'opportunité d'utiliser les outils de datamining dans l'exploration de l'entrepôt afin de mieux comprendre les patients de l'hôpital, les soins prodigués, le cheminement des patients et bien d'autres questions toutes aussi naturelles que difficiles. Vu le volume phénoménal des données entreposées, et l'état expérimental de notre démarche, nous devons cibler, dans une première étape, un groupe restreint de patients et de services. C'est pourquoi, nous nous contentons des malades admis au service de cardiologie pour une première période d'une année, soit toute l'année 2002. Notre travail doit répondre à deux questions essentielles. La première est de vérifier si les données telles que stockées dans l'entrepôt de données, sont d'une qualité qui permet de les exploiter directement par les outils de datamining. Si la réponse est non, quels sont les moyens à mettre en œuvre pour rendre ces données exploitables. La deuxième question est comment caractériser un patient ? Ou plus clairement, comment décrire la visite d'un patient ? Pour cela, il nous fallait impérativement répondre à de nombreuses interrogations. Existe-t-il des catégories de patients ? Par quoi sont-il caractérisées et comment les identifier? Comment affecter un patient à une ou plusieurs catégories ? Quel a été le cheminement du patient lors d'une visite? Est-il le cheminement optimal? Quelle aura été l'issue de sa visite en terme de guérison, temps de séjour, efficacité du traitement s'il avait emprunté un autre cheminement? Est-il possible d'estimer le diagnostic principal d'un patient en fonction des données recueillies (personnelles, tests,...)? Ou est ce que le diagnostic dépend d'autres paramètres que le système d'information ne couvre pas ou qui sont trop complexes et subjectifs pour être capturés par un quelconque système? Ceux-ci sont des exemples des questions fondamentales auxquelles nous souhaitons répondre dans notre travail.

Afin de mener à bien notre travail, nous devons étudier toutes les méthodes existantes et faire une comparaison entre elles pour, en première étape, déterminer leurs caractéristiques (types de données qu'elles prennent en charge, le volume de données, le type de modèle en sortie, les performances de traitement,...) et ensuite faire le choix, à la lumière des résultats de comparaison, des méthodes et techniques à utiliser.

[1] CIRESSS : Terme utilisé par abus de langage pour désigner l'entrepôt de données du CHUS, créé par la société SAND sous sa technologie Nucleus dans le cadre du projet CIRESSS (Centre informatisé de recherche évaluative en services et soins de santé)

Une analyse rapide des données permet d'aborder le problème de la qualité des données. Bien que la plupart des données collectées au CHUS sont capturées automatiquement par le système informatique (ex : pression artérielle, tests de laboratoire, tests cardiaque...), les cas d'omission et d'erreur sont fréquents. Aussi, la multiplicité des points de saisie de données a généré une diversité de format pour des données de même type. Un travail préliminaire consiste forcément en une redéfinition plus rigoureuse des types de données et une transformation des données de même type en un seul format standard. L'étude de certaines exceptions qu'on a retrouvées telles que les cut-off, les valeurs limites,... (Ex: <0.1, analyse impossible, ...) s'avère aussi déterminante. La bonne compréhension d'une exception peut à elle seule orienter grandement la détermination d'un diagnostic ou l'abandon d'une piste d'investigation.

Peut on établir un diagnostic en fonction des données recueillies? C'est la première question qui vient à l'esprit de tout un chacun. Question que nous avons transformée en la suivante. Est ce que la description du cheminement d'un patient est fortement caractérisée par son diagnostic final ? Pour y répondre, nous avons décidé d'appliquer le clustering. Si les patients seront regroupés en fonction de leurs diagnostics principaux, la réponse est affirmative. Dans le cas contraire, notre hypothèse est fausse.

Le choix des données à prendre en compte pour établir un modèle s'avère des plus importants. Dans la littérature rare sont les études qui utilisent plus de 40 paramètres (personnelles, signes de vie, antécédents, paramètres de risque, tests,...). Cela est dû à l'interdépendance entre les paramètres et leur non disponibilité pour toutes les visites. Nous avons à effectuer avec l'aide des spécialistes du domaine (docteurs et biochimistes) une sélection draconienne des données pour ne laisser que celles déterminantes pour l'établissement d'un diagnostic en cardiologie et pour lesquelles les omissions et les erreurs ne sont pas paralysantes. Le sexe, l'âge, l'état de santé, l'état d'avancement de la maladie, certains signes de prédisposition, le choix des types et moyens de tests,... font que la prise en charge peut être différente d'un patient à un autre pour un même et unique diagnostic principal. De ce fait, la détection des sous populations s'avère difficile. Les éléments discriminatoires sont noyés dans la complexité de la pratique médicale. Les techniques de clustering dont les résultats dépendent grandement des paramètres initiaux (fonction de distance, cut-off, poids,...) sont difficiles à mettre en œuvre. Proposer une méthode qui compose avec toutes les difficultés énumérées, pour répondre à notre question, est notre défi.

3. ENVIRONNEMENTS DE TRAVAIL

3.1. Banque de données

Le terme banque de données est souvent confondu avec base de données dans l'esprit même des informaticiens. Le grand dictionnaire du Québec en donne cette définition :

> *«Ensemble d'informations organisées autour d'un même sujet, directement exploitables et proposées en consultation aux utilisateurs.»*[2]

La notion de banque de données représente l'accumulation de renseignements sous une version électronique indépendamment de la structure qui supporte ces renseignements et de la représentation qu'ils peuvent prendre. C'est donc, un ensemble d'informations relatives à un domaine défini de connaissances et organisées pour être accessible par plusieurs utilisateurs. Le but principal de la création d'une banque de données est de mettre à la disposition de l'utilisateur, expert du domaine ou novice à ses débuts d'apprentissage, une collection d'informations qu'il pourra consulter grâce à des outils qu'elle lui offre et qui lui permettent de dégager très rapidement la documentation sur un sujet déterminé. Pour éclaircir la confusion qui existe entre base de données et banque de données, on peut d'ores et déjà dire, avant même la définition de la notion de base de données que cette dernière est souvent l'instrument informatique, nécessaire, destiné à recevoir les informations qui constituent une banque de données. Une banque de données regroupe souvent plusieurs bases de données.

Une banque de données est caractérisée par :

- La structure informatique qui la supporte
- Les modes d'accès à l'information qu'elle offre
- Les modes de représentation de l'information retournée

Vu qu'une banque de données peut contenir un volume gigantesque d'information, il est important pour l'utilisateur qu'elle lui offre des outils efficaces qui le guident et lui permettent d'effectuer aisément ses recherches.

Moteur de recherche

Un moteur de recherche est souvent nécessaire dans une banque de données afin d'aider l'utilisateur à repérer rapidement les domaines d'intérêts ciblés. La recherche se fait souvent par concepts appelés aussi éléments-clés de la banque de données ou par mots-clés sachant que chaque concept est relié à un ou plusieurs mots-clé. La bonne pratique de la limitation de la recherche peut s'avérer utile si l'information retournée est très grande. La limitation est recommandée pour une approche de recherche préliminaire. Elle peut se faire en spécifiant un nombre maximal d'objets à retourner, en précisant des conditions de recherche restrictives ou, encore, en ciblant une partie de la banque de données.

[2] www.granddictionnaire.com : Dictionnaire terminologique en ligne de l'Office québécois de la langue française

Requête simple

Devant l'énormité du volume des données et l'affût d'utilisateurs de divers horizons avec la mise à disposition de banques de données sur le Web, de plus en plus de banques de données s'équipent d'un éditeur de requêtes simples et proches du langage naturel grâce auquel l'utilisateur n'a pas à faire l'effort d'apprendre un langage formel d'interrogation de base de données ni de connaître la structure interne de la banque. Toutefois, l'utilisateur n'est pas libre de formuler sa requête comme il veut; il doit respecter un minimum de structures de phrases et avoir une bonne compréhension des concepts clés stockés dans la banque.

Navigateur (Browser)

Conçu pour les banques de données peu volumineuses, le navigateur reste assez efficace mais son utilisation devient très vite fastidieuse avec l'augmentation de la taille des données consultées. L'utilisateur peut, alors, très facilement se perdre dans l'arborescence des détails. Combiné à un moteur de recherche ou une requête, il permet de prospecter en large après avoir repéré l'information utile.

3.2. Base de données

Basée sur une théorie fondamentale bien établie et sur des techniques de modélisation et des algorithmes de traitement des données, la science des bases de données est une discipline sérieuse qui est née essentiellement pour répondre aux besoins de la gestion. Les systèmes de gestion de bases de données (SGBD) sont l'aboutissement de la recherche effectuée dans ce domaine. Ils permettent une gestion efficace des bases de données. Par abus de langage le terme de base de données est utilisé à tort et à travers pour désigner tout ensemble de données plus ou moins structurées. Alors qu'une base de données est un ensemble de données qui satisfait les conditions suivantes [GARBD99]:

- *Ses données sont interdépendantes*
- *Il modélise une partie du monde réel*
- *Il a pour but final de supporter une application informatique*
- *Il est interrogeable par le contenu pour déterminer et extraire tous les objets qui satisfont à certains critères*
- *Ses objets ont une structure qui peut être explorée.*

Système de Gestion de Base de Données (SGBD)

La notion de base de données ne peut pas exister si elle n'est pas dotée d'un SGBD. Un SGBD est une collection de logiciels dont la mission est de permettre à plusieurs utilisateurs de partager efficacement une base de données dans le but d'y insérer, modifier, supprimer et rechercher des données spécifiques selon un ou plusieurs critères en relation avec les données et définis par l'utilisateur. L'efficacité d'un SGBD est vérifiée par sa capacité à rester transparent à l'utilisateur final, sa gestion des données partagées entre les utilisateurs que ce soit pour consultation ou pour modification et la vitesse de traitement des requêtes de l'utilisateur. En plus de ces fonctions indispensables à un SGBD, celui-ci doit

assurer la sécurité et l'intégrité des données dans un environnement multi-utilisateurs. Avant d'aller plus loin, nous définissons quelques notions et concepts manipulés par les SGBD.

Système de Gestion de Fichiers (SGF)

Un système de gestion de fichiers est un composant de plus bas niveau qu'un SGBD. Souvent le SGBD est construit au-dessus d'un SGF, tout en offrant des fonctionnalités supplémentaires telles que:

- *La description des données:* définition du type, du format, des caractéristiques et éventuellement d'une fonction de calcul séparément de l'utilisation
- *La gestion de la structure des données:* capacité de reconnaître la structure d'une donnée à partir du nom de la donnée

3.2.1. Types de bases de données

Le modèle réseau

Les bases de données réseaux et hiérarchiques, appelées aussi modèles d'accès parce qu'elles privilégient l'optimisation des entrées-sorties, sont des modèles dans lesquels une base de données est un ensemble d'articles organisés en fichiers reliés entre eux par des pointeurs et offrant un langage de navigation dans le but de parcourir et consulter les fichiers.

Les modèles réseaux ont vu leur apparition au tout début des années 70. Ils manipulent trois concepts clés que sont l'atome, le groupe et l'article et des relations entre articles appelées associations.

Le modèle hiérarchique

Modèle plus efficace et plus répandu que le modèle réseau. Les objets qui le définissent sont :

Champs : l'équivalent de l'atome dans le modèle réseau, donc, c'est la plus petite information élémentaire ayant un nom et un type.

Segment : Ensemble de champs pouvant être rangés de façon séquentielle dans la base de données possédant un nom et constituant l'unité d'échange entre la base de données et les applications.

Arbre de segments : Ensemble de segments reliés par des liens père-fils, constituant de ce fait une hiérarchie.

Chaque type de segment père possède N1 types de segments fils et chaque occurrence d'un type de segment père possède Nx occurrences de chaque type de segment fils. Chaque type de segment fils peut être considéré comme une racine d'une autre relation père-fils, jusqu'à atteindre des segments qui n'ont pas de fils, appelé segment feuille. Le graphe hiérarchique résultant de cette organisation d'arbres est appelé forêt.

Contrairement au modèle réseau, les segments ne peuvent avoir des données répétitives. Certains modèles hiérarchiques autorisent des liens horizontaux entre segments afin de réduire les duplications et étendre les possibilités du modèle.

Le modèle relationnel

Proposé par E.E. Codd en 1970, le modèle relationnel est basé sur la théorie des ensembles et est très simple à comprendre et à modéliser. Ses concepts et structures de donnée de base reposent sur et rappellent les termes mathématiques correspondant. Les objets du modèle relationnel sont :

Domaine: Un domaine de valeurs est l'ensemble où une donnée prend sa valeur. Les entiers, réels, caractères sont des exemples de domaines de valeurs.

Relation : Pour définir une relation, il nous faut définir le produit cartésien de domaines.

Produit cartésien de domaines: Soit D1,D2,...,Dn des domaines de valeurs, le produit cartésien des domaines D1,D2,...,Dn est l'ensemble des vecteurs (V1, V2, ..., Vn) où Vi ($i \in [1,n]$) est une valeur du domaine Di

Une relation est un sous-ensemble d'un produit cartésien d'un ensemble de domaines et possédant un nom. Elle est représentée par une table à deux dimensions où les lignes représentent les vecteurs et les colonnes représentent les domaines, sachant qu'un domaine peut se répéter plusieurs fois.

Attribut : Colonne d'une relation possédant un nom et prenant ses valeurs dans un ensemble fini de valeurs appartenant au domaine dans lequel est défini la colonne.

Tuple: Ligne d'une relation prenant ses valeurs dans l'ensemble des vecteurs V1,V2,..., Vn

Un algèbre relationnel a été inventé par E.E.Codd afin de manipuler les relations. Les noms de ses opérations rappellent celles des mathématiques. On peut en citer l'union, la différence, le produit cartésien, la projection, la restriction, la jointure, l'intersection, la division. D'autres opérations ont été définies par des chercheurs mais peuvent toutes être dérivées des opérations de base.

Le modèle objet

Une base de données objet (BDO) est tout d'abord un SGBD qui fournit toutes les fonctionnalités nécessaires. Contrairement à la plupart des langages orientés objet tel que C++ ou Java, un objet d'une base de données orientée objet doit être *persistant*, c'est-à-dire que sa durée de vie ne se termine pas avec la fin du programme qui le manipule [ATKINSON89]. Il doit donc pouvoir être stocké sur disque.

Un objet dans une BDO peut aussi avoir une *version*. La gestion des versions d'un objet permet de remonter les modifications successives apportées à celui-ci dans les deux sens. Un objet qui permet la gestion de ses versions est appelé objet versionnable. En plus, tout objet d'une BDO doit vérifier des propriétés orientées objet telles que l'identité, l'héritage simple et multiple, le polymorphisme, les messages d'exception,...

Un objet interagit forcément avec d'autres objets. Il a souvent besoin d'un pointeur qui concrétisera cette relation. La sauvegarde et la restauration de l'objet et de son pointeur sur disque est un des problèmes que le SGBD a à traiter. Pour cela il existe plusieurs solutions dont La persistance par héritage et la persistance par référence [ATKINSON00].

3.3. Bases de données décisionnelles

L'exploitation des données informatiques générées par les bases de données transactionnelles dans un but autre que celui pour lequel elles ont été développées a toujours posée problème. Pour l'analyse des données, on a souvent recours aux statistiques pour les synthétiser et les présenter sous forme de tables résumés facile à comprendre et à interpréter et surtout plus utiles pour l'analyse. Malheureusement, les calculs statistiques sont gourmands en temps de calcul et en accès disque, puisqu'il est souvent nécessaire de parcourir plusieurs fois toute la base de données pour ressortir l'information voulue, ce qui peut constituer un obstacle réel au bon déroulement des affaires courantes. Aussi, l'architecture interne de la base de données, comme elle contribue grandement à augmenter et optimiser les performances générales du système pour les besoins de production, elle peut aussi être la raison principale de la médiocrité des performances des outils d'analyse. C'est essentiellement pour ces deux raisons que les bases de données décisionnelles ont vu le jour.

Les infocentres ont été, au début des années 80, la première réponse aux grandes différences qui existaient entre les systèmes de production OLTP (On Line Transaction Processing) et les systèmes de décision OLAP (On Line Analysis Processing). C'est une technologie consistant à récupérer des parties de Base de données sur une seule machine pour les traitements décisionnels. C'est le premier genre de base de données type entrepôt de données. Le principe des infocentres est d'intégrer des vues de multiples sources de données et de les exploiter pour le décisionnel.

En suite, d'autres systèmes plus performants ont vu le jour. Les middlewares et plus particulièrement les entrepôts de données ont été la solution et sont aujourd'hui très répandus.

> ***Le middleware:*** est un ensemble d'outils informatiques installés sur un seul poste client à partir duquel on peut accéder à toutes les données de l'entreprise par la technique du *Data Pull*.
>
> ***L'entrepôt de données:*** est un ensemble de données historisées constitué par extraction à partir de bases ou de fichiers, organisé par sujets, consolidé dans une base de donnée unique, géré dans un environnement de stockage particulier, et aidant à la prise de décision dans une entreprise.

Nous étudierons plus en détails les entrepôts de données et les bases de données décisionnelles dans le troisième chapitre dans lequel nous aborderons les techniques d'extraction, de fusion et d'exploration des données, nous expliquerons la nécessité de la modélisation par sujet et décrirons dans les détails le kit

d'alimentation d'un entrepôt de données. Nous expliciterons aussi la notion de cube multidimensionnel en précisant ses apports et ses inconvénients.

4. CONCLUSION

Pour introduire notre travail, nous avons commencé par lever certaines ambiguïtés relatives au domaine de la connaissance. Nous avons commencé par définir les notions de base sur lesquelles repose notre travail telles que la donnée, l'information, la connaissance et le savoir. Nous avons relevé l'inadéquation des systèmes de stockage traditionnels aux exigences que requiert l'extraction de la connaissance et l'évolution de ceux-ci vers des systèmes plus appropriés tels que les bases de connaissances et les entrepôts de données.

En plus de la définition de la problématique, nous avons défini, dans ce premier chapitre, les types d'environnements dans lesquels peuvent se dérouler le genre de travail que nous avons à effectuer. Trois générations de base de données se sont succédées. Les modèles d'accès que sont le modèle réseaux et hiérarchique, ont été introduits dans les années 70. La deuxième génération a vu l'apparition du relationnel dans les années 80. Puis, vers le début des années 90, est apparu l'orienté objet qui s'est moins imposé comme révolution que comme continuation par enrichissement du relationnel.

Les enjeux futurs des SGBD sont la prise en charge de quantités énormes de données, surtout celles provenant du WEB en permettant la recherche par le contenu, la gestion des bibliothèques multimédia et aussi l'intégration du décisionnel en parallèle avec le transactionnel.

Les entrepôts de données restent relativement jeunes comme technologie et constituent un nouveau marché en pleine expansion. La plupart des constructeurs ont intégré des outils de création et de gestion des entrepôts de données. Ils développent aussi des outils d'analyse qui sont de plus en plus performant faisant appel à la statistique, aux techniques de fouille de données et à des représentations graphiques trois dimensions.

5. PLAN DU MEMOIRE

Notre mémoire est divisé en trois parties fondamentales. Dans la première partie, constituée du chapitre deux, nous avons commencé par décrire les phases nécessaires à l'extraction de la connaissance, tout en axant notre analyse sur le besoin de standardisation des étapes de datamining, l'épineuse question d'éthique ainsi que sur d'autres aspects relatifs au techniques de datamining. Nous avons, ensuite, défini et décrit plusieurs techniques de datamining en prenant soin de préciser les éléments de différence et de comparaison. Le chapitre trois représente la deuxième partie de notre travail, laquelle commence par décrire les entrepôts de données de façon générale, pour se focaliser, par la suite, sur Nucleus Server, l'entrepôt de données sur lequel nous avons travaillé et qui repose sur une technologie innovatrice basée sur la tokenisation, l'architecture orientée colonne et la compression des données. La troisième partie de notre mémoire, représentée par le chapitre quatre, traite du travail réalisé et des contributions apportées. Nous y expliquons les différentes étapes de prétraitement des données avant l'application de notre méta-

méthode. Nous y expliquons et défendons nos différents choix. Nous y décrivons, aussi, notre méta-méthode, les résultats obtenus et les améliorations apportées. Nous terminons notre mémoire par une conclusion et les perfectionnements et perspectives que nous envisageons.

Chapitre II

L'EXTRACTION DE LA CONNAISSANCE

1. EXTRACTION DE LA CONNAISSANCE

La datamining ou forage de données est l'ensemble des techniques permettant d'extraire de la connaissance, sous forme de modèles, à partir des bases de données historisées, afin de décrire et prédire le comportement d'un procédé. Il est important de faire la différence entre le datamining et le knowledge discovery (KD), ou l'extraction de connaissances. Dans beaucoup d'ouvrages, les termes datamining et de knowledge discovery sont considérés comme équivalents. U. Feyyad [Fayyad96] a donné des définitions assez générales pour contenir des deux termes mais assez précises pour les dissocier.

> *Le datamining est un processus non-trivial d'identification de structures inconnues, valides et potentiellement exploitables dans les bases de données*

Alors que

> *le* knowledge discovery *est un processus qui utilise les méthodes du datamining (algorithmes) pour extraire (identifier) les connaissances enfouies selon des spécifications de mesures et de seuils, à partir de données sur lesquelles on a menée d'éventuelles opérations de prétraitement, d'échantillonnage et de transformations.*

Bien que générales ces définitions ont l'avantage de couvrir relativement bien toutes les activités relatives au KD. L'auteur revient pour redéfinir le datamining comme suit:

> *Le datamining est une étape du processus du KD qui se constitue d'algorithmes particuliers de fouille de données, lesquels produisent, avec certaines limitations informatiques acceptables et efficaces, des patrons particuliers représentant les données.*

Généralement, et par abus de langage, on utilise le terme datamining pour parler de KD. Il faut pouvoir situer le datamining, et le considérer comme une simple étape du KD. Alors que ce dernier est un processus plus global qui se divise en quatre grandes phases qui sont l'entreposage des données dans des entrepôts de données (DataWarehousing), le pré-traitement (pré-processing), le forage des données (DataMining) et enfin le post-traitement (post-processing). La phase de l'entreposage des données a pour but d'alléger la charge sur les bases de données transactionnelles et aussi de centraliser l'information concernant un ou plusieurs sujets les plus susceptibles de faire l'objet d'analyses futures. La phase de prétraitement des données consiste en plusieurs opérations incluant le traitement des données manquantes, la correction des données erronées et la transformation des données. La phase post-traitement a pour but l'interprétation des patrons de données découverts. Les quatre phases font parties de tout un processus itératif d'extraction de connaissances et complètent, certaines en amont et d'autres en aval, la phase du datamining qui reste le cœur du KD.

Le KD est un processus itératif dans lequel les différentes phases peuvent se chevaucher et dont les résultats d'une itération peuvent être utilisés comme entrée de la prochaine itération. Les sorties d'une

phase sont utilisées comme entrées de la phase suivante ou comme feed-back aux étapes précédentes dans le but d'améliorer les résultats de la prochaine itération.

Les patrons ou modèles représentant les connaissances extraites peuvent être de deux types différents. Les modèles fonctionnels comme par exemple la distribution moyenne du trafic en fonction des heures de la journée, et les modèles logiques comme "si facture > 100.000 $ et facture impayée alors faillite à 70%". Pour pouvoir représenter les situations complexes du monde réel, il est parfois utile d'utiliser des modèles hybrides conjuguant les modèles fonctionnels et les modèles logiques.

1.1. La phase de prétraitement

Elle se compose de trois grandes étapes, le nettoyage des données, la transformation des données et la sélection des attributs. Les étapes ne sont pas nécessairement exécutées dans cet ordre. Le nettoyage des données inclut la vérification de la qualité et la cohérence des données, la correction des erreurs et des données erronées, le remplacement ou la suppression des valeurs manquantes, la suppression des valeurs extrêmes et excentrées, etc. Elle peut aussi contenir les opérations d'échantillonnage et de transformation des données, comme par exemple l'ajout de données extraites du Web ou d'une autre source de données, le calcul de sommes, moyennes,... Les tâches des trois étapes sont effectuées partiellement lors de la construction de l'entrepôt de données, seulement, il est toujours nécessaire de refaire ces mêmes opérations dans le but de raffiner les données et permettre une meilleure maîtrise du problème. Il arrive aussi qu'une méthode de forage de données ait besoin de données formatées d'une façon particulière. Les méthodes de nettoyage sont inhérentes au domaine d'étude et la participation de l'utilisateur final est souvent cruciale. Quoique le processus du KD offre des moyens de détection des données contenant du bruit, il peut exister des données atypiques et correctes en même temps. Le choix de l'élimination de telles valeurs revient à l'utilisateur en prenant en compte le type de travail à réaliser et les objectifs à atteindre. L'identification des erreurs et des valeurs excentriques peut se faire grâce aux techniques de visualisation graphiques [Simoudis96]. La préparation des données représente jusqu'à 60% du temps total du processus de datamining [Cabena98], [Witten00]. Dans le cas des grandes bases de données, la sélection des données, le choix des attributs pertinents pour l'étude ainsi que leurs discrétisations sont d'une importance capitale.

1.2. La phase de datamining

La phase de datamining consiste en l'utilisation d'une ou de plusieurs techniques appliquées aux données, suivi de l'interprétation des résultats. Une phase de post-traitement peut être requise. Bien que la phase de forage des données est la phase la plus importante du processus de KD, rappelons que cette phase ne représente que 20% du temps de tout le processus [Fayyad96]. Elle est aussi la phase la plus étudiée par les chercheurs parce que, d'une part, les autres phases sont très peu automatisables et d'autre par, elle est la raison d'être des autres étapes et constitue un enjeu majeur d'une nouvelle nature.

1.3. La phase de post-traitement

Les résultats en sortie de l'algorithme de forage de données peuvent être raffinées dans la phase de post-traitement. Celle ci peut se résumer dans certains cas à l'interprétation des connaissances découvertes ou en des traitements supplémentaires sur les connaissances extraites. Il arrive que le post-traitement soit directement inclut dans la phase de datamining, mais il est préférable de séparer les deux phases. Le but ultime de cette phase est de permettre une meilleure compréhension et interprétation des connaissances générées par les algorithmes de forage de données. Le moyen le plus simple et le plus efficace, dans la plupart du temps, pour faire cela est le recourt aux techniques de visualisation graphique. D'autres techniques ont été spécialement développées pour certains algorithmes de forage de données. Des propositions ont été faites pour, par exemple, déduire à partir des poids des interconnexions d'un réseau de neurones des règles d'induction [Schetinin96], [Andrews95]. Aussi, le raffinement des règles associatives issues de la technique du même nom, en supprimant les règles rares, les règles triviales, les règles obsolètes et les règles particulières est une autre forme de post-traitement [Pasquier00].

1.4. Les modèles

La découverte de modèles, qui est le but principale du datamining, consiste à trouver à partir de bases de données des modèles de description à partir des quels on pourrait éventuellement faire la prédiction. À chaque modèle, on peut associer un indicateur de confiance qui représente la probabilité que le modèle soit vérifié. Il existe deux types de modèles que sont :

Modèle fonctionnel :

> *Technique permettant d'approcher une dépendance entre N variables X1, X2,...Xn d'entrée et une variable de sortie Y par une fonction Y=F(X1, X2,...Xn) avec un éventuel indicateur de confiance.* [Gardarinii99]

Modèle logique :

> *Technique permettant d'exprimer des relations entre données par un ensemble de règles du type «si X alors Y», avec un éventuel indicateur de confiance.* [Gardarinii99]

2. STANDARDISATION DES ETAPES DE DATA MINING

Il est difficile de donner une définition exacte et complète de la phase de datamining, une définition qui capturerait les grands aspects et les détails de la phase, car la modélisation, la comparaison, l'évaluation de la connaissance extraite par le moyen du datamining, faute de maturité peut-être, ne sont pas encore standardisés. Des efforts de standardisation sont en cours. CRISP-datamining[5] (CRoss Industry Standard Process of datamining) en est un exemple.

Le modèle du CRISP-datamining est comme représenté à la *figure 02*.

[5] www.crisp-dm.org

- **La phase de compréhension du domaine :** Elle s'intéresse à la compréhension des objectifs et besoins pour les convertir en un problème de datamining. Un plan préliminaire est établi pour atteindre les objectifs tracés. Elle se fait en quatre étapes, la détermination des objectifs, l'évaluation de la situation en termes de ressources (contraintes, hypothèses et autres), détermination des buts du datamining qui consiste en la conversion des objectifs généraux en buts techniques à réaliser par les outils de datamining et enfin la production d'un plan de projet pour réaliser les objectifs. Les étapes du projet y sont décrites avec les durées, les ressources requises, les entrées et les sorties ainsi que les dépendances.
- **La phase de compréhension des données :** Elle commence par un ensemble initial représentatif des données dans le but de se familiariser avec ces dernières, de découvrir les problèmes liés à leur qualité et de repérer des sous-ensembles de données potentiels pour de plus amples analyses. Elle se fait en quatre étapes, la collecte des données initiales, la description des données, la vérification de la qualité des données et l'exploration des données.

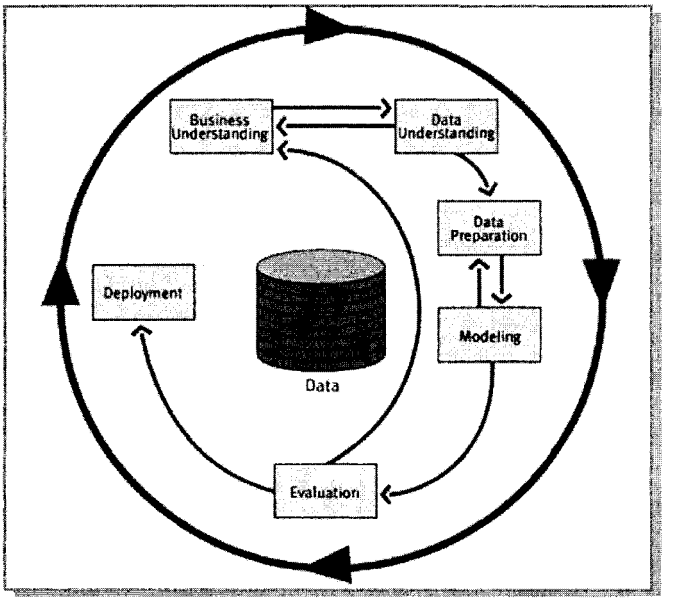

Figure 02 : Le modèle de CRISP-datamining du processus d'extraction de connaissances «Source www. crisp-dm.org»

- **La phase de préparation des données :** Elle couvre toutes les activités pour construire l'ensemble des données. La préparation des données peut apparaître plusieurs fois à tous les niveaux sans un ordre précis. Elle inclut la sélection des tables, des attributs et des enregistrements, la transformation et le nettoyage des données. Elle se fait en cinq étapes, la sélection des données, le nettoyage des données, la construction des données en générant

des données dérivées, en créant de nouveaux enregistrements et surtout en transformant les données existantes, l'intégration des données à partir de plusieurs tables et le formatage des données qui consiste généralement en des modifications syntaxiques qui ne changent pas le sens des données mais restent nécessaires pour les algorithmes de datamining.

- **La phase de modélisation :** Elle consiste en la sélection et application des techniques de datamining en ajustant leurs paramètres pour des résultats optimaux. Les techniques sont différentes et requièrent souvent des paramétrages différents et des données en entrée différentes. L'interaction avec la phase de préparation des données est très grande. Elle est composée de quatre étapes, la sélection des techniques de modélisation les plus susceptibles de satisfaire les objectifs, la détermination d'une méthode de tests qui se résume souvent en la séparation des données en un ensemble d'entraînement et un ensemble de tests, la construction du modèle et enfin l'évaluation du modèle selon les critères techniques de datamining.
- **La phase d'évaluation :** Elle consiste en l'évaluation des modèles choisis et appliqués aux données. Cette évaluation porte sur l'application et l'apport des modèles sur la réalité et non sur les performances des modèles du point de vue datamining, évaluation qui se fait lors de la phase précédente. La revue des étapes de construction des modèles est aussi effectuée. L'évaluation se compose de trois étapes, l'évaluation des résultats en fonction des objectifs établis, la revue du processus pour une éventuelle remise en cause de la démarche suivie et la détermination des prochaines étapes en optant soit pour un déploiement ou pour d'autres itérations dans le but d'améliorer les résultats ou extraire d'autres sur la lumières des précédents.
- **Phase de déploiement :** Cette phase dépend des objectifs. Elle peut être aussi simple que la génération d'un rapport ou aussi complexe que l'intégration de la connaissance extraite aux données de départ pour construire un processus itératif complet. Elle se compose de quatre étapes, l'élaboration d'un plan de déploiement, l'élaboration d'un plan de contrôle et de maintenance, la production d'un rapport final et enfin la revue et l'évaluation de tout le projet.

3. DATA MINING ET ETHIQUE

Il faut faire très attention à la question de l'éthique dans la manipulation des données personnelles ou privées. Le domaine médicale est sans doute le plus concerné par cet aspect. Dans le domaine bancaire, les outils de datamining sont souvent utilisés pour aider les décideurs à accorder ou pas un prêt bancaire à un client et estimer le montant maximal à lui octroyer. Ils sont aussi utilisés pour sélectionner les personnes à cibler par une campagne publicitaire. Les réponses à ce type de questions sont basées sur les données personnelles des clients. Elles utilisent surtout des discriminations justifiées et nécessaires telles que la race, le sexe, la religion et la classe sociale. Ces discriminations sont inadmissibles du point de vue éthique et illégales du point de vue juridique. Dans un contexte médical, prendre en considération

le sexe et la race pour établir un diagnostic médical est par contre éthique et même souhaitable pour le malade concerné.

Enlever automatiquement les données sensibles lors de la construction de modèles de datamining n'est pas suffisant. Il arrive que d'autres données, a priori sans risque, apportent des nuances lesquelles mises ensemble forment une information dont l'utilisation est condamnable du point de vue éthique. On peut citer, pour illustrer nos propos, certains tests médicaux qui ne s'appliquent qu'aux femmes, le code postal par lequel on peut cibler des communautés se regroupant dans des villes ou des quartiers précis, ces regroupements peuvent être de type racial, social ou autre, le nom de la personne qui peut être de connotation asiatique, européenne de l'est, arabe,...

Il est généralement admis que toute collecte de données personnelles passe par le consentement des concernés. Des explications claires, en des termes qu'ils peuvent comprendre doivent leurs être fournis pour répondre à des questions essentielles telles que comment et pour quel but sont utilisées les données, quelles sont les mesures prises pour la protection de la confidentialité et l'intégrité des données, quelles sont les conséquences d'une utilisation malhonnête des données et quels sont les recours offerts pour y remédier. Malheureusement dans beaucoup de cas et notamment dans le domaine médical, les données sur lesquelles on veut travailler ont été collectées et saisies pour d'autres buts que le datamining, ce qui rend la situation plus délicate. La propriété des données doit être précisée clairement (hôpital, Régis, gouvernement,...) et les responsabilités qui en découlent doivent être définies. La possession des données ne signifie nullement pouvoir les utiliser sans aucune restriction.

L'analyse des données peut aussi révéler des découvertes très inattendues et parfois carrément étranges. Une étude en France a montré que les propriétaires de voitures de couleur rouge sont plus susceptibles d'être en défaut de payement de leurs prêts voiture [Witten00]. Sans trop surenchérir sur la véracité d'une telle découverte, sur le type d'information utilisée et où cette information a été collectée, est-il éthique de prendre en considération une telle découverte qui discrimine toute personne qui possède une voiture rouge? Le bon sens dirait non, les banques qui défendent leurs intérêts considéreront sûrement cette information autrement.

Il faut toujours déterminer qui a accès aux données, pour quel but et quels sont les types de conclusions qu'il pourrait émettre en analysant les données. Pour cela, il est important de considérer les us et coutumes de la communauté étudiée, des standards mis en place et le cadre légal qui régit le domaine d'étude et l'utilisation des données. Les ordonnances des médecins, même dénominalisées, peuvent elles être accessibles à tout le monde? Si l'hôpital décide de le faire, dans le but noble d'instaurer une meilleure pratique médicale, il sera peut être la cible de pressions de certains groupes pharmaceutiques qui subventionnent des projets de recherches. Pour prévenir ce genre de dérive, la communauté scientifique se doit d'établir ses propres standards, en plus de ceux mis en place, qui accompagneront les processus d'accès aux données, d'analyse et d'interprétation des résultats.

Une conclusion, fut elle le résultat d'une analyse statistique correcte, doit toujours être accompagnée de remarques et d'avertissements sur son utilisation, car le but final du processus d'exploration des données et d'extraction de la connaissance n'est pas la prise de décision mais l'apport d'éléments, lesquels associés à d'autres plus logiques et plus éprouvés, permettront des décisions plus clairvoyantes.

Les outils de datamining posent un autre problème d'ordre politique. L'exemple de la vente de détail est édifiant. Sachant que le consommateur achète souvent les liqueurs et les chips, faut-il mettre les deux produits sur la même rangée ou faut-il les séparer pour obliger le consommateur à passer plus de temps dans le magasin ? Faut il mettre près d'eux les produits les plus générateurs de profits ou les produits dont la date de péremption est proche ? Bien que l'incitation à la consommation, même inutile, constitue le moteur de l'économie, il n'en demeure pas moins que cette question reste entière pour la discipline du datamining.

Les standards imposés par la communauté, les standards scientifiques, la sagesse et le bon sens dans l'intérêt de la communauté des décideurs restent à ce jour les seules moyens qui régissent l'éthique du datamining. Avec la multiplication de la quantité de données, la disponibilité des moyens technologiques et les concurrences féroces dans tous les domaines, il est temps de penser plus sérieusement aux outils de datamining et leurs implications éthiques et juridiques.

4. LES VALEURS MANQUANTES

En pratique, il n'existe presque pas de données complètes. Les données manquantes, indépendamment des causes qui les ont générées, font parties intégrantes et indissociables des données. Généralement, on remplace les données manquantes par des données fictives qui n'appartiennent pas au domaine des valeurs des variables concernées; des valeurs négatives pour les données numériques qui sont toujours positives, zéro pour celles qui ne sont jamais nulles. Pour les valeurs nominales, un blanc ou un tiret peut être utilisé. Il arrive qu'on ait différents types de valeurs manquantes pour une même variable. Une valeur manquante peut être inconnue, non enregistrée ou incorrecte. Le traitement des différents types de valeurs manquantes peut se faire par leurs remplacements par des valeurs négatives (-1, -2, -3,...) quand celles-ci n'appartiennent pas au domaine de valeur de la variable.

Il faut penser attentivement aux significations des valeurs manquantes. Elles peuvent apparaître pour différentes raisons comme le disfonctionnement d'un appareil de mesure dans le domaine médicale, le changement de la méthode dans un processus de collecte de données, le refus de répondre à certaines questions dans un sondage, l'impossibilité pratique de mesurer la taille d'un animal sauvage dans une expédition scientifique, la mort prématurée ou la disparition d'un cobaye avant la fin d'une expérience de laboratoire. La détermination de la nature de la donnée manquante peut être interprétée et prise en compte dans l'élaboration du modèle de datamining.

Certains algorithmes de datamining ne donnent aucune signification aux données manquantes et les considèrent seulement comme pas connues. Il peut exister une bonne raison qui a fait que la valeur est manquante, comme par exemple l'inutilité de faire un test médical qui s'avérerait non interprétable dans

une situation donnée. Dans ce cas, la valeur manquante renseignerait sur un état précis du patient en plus du fait qu'elle soit manquante. Remplacer la valeur manquante par une valeur significative serait souhaitable, mais le choix de la valeur de remplacement est des plus délicat. La participation des experts du domaine est recommandée, voir même indispensable. Dans le domaine médical, il arrive que l'interprétation de la valeur manquante d'un test, dans un contexte particulier, suffise à elle seule pour établir un diagnostic.

5. LES VALEURS ERRONÉES

Certains champs de valeurs qui n'avaient pas un grand intérêt lors des différentes mises à jour de la base de données, ont été soit laissés vide, soit leurs contenu n'a pas fait l'objet d'un contrôle strict. Comme leurs contenus n'affectaient pas le but initial pour lequel elles avaient été collectées, elles n'ont fait l'objet d'aucun changement après avoir constaté les erreurs qu'elles recelaient, ni de correction après vérification postérieure. Le contenu de telles variables peut prendre une grande importance pour les algorithmes de datamining. Une valeur omise ou erronée peut grandement affectée les performances d'un modèle, voir même fausser les résultats. Dans le domaine médical, l'emploi d'un patient n'est pas une donnée très pertinente pour son traitement médical, mais peut le devenir dans un processus d'extraction de connaissance, si on s'intéresse à analyser une maladie susceptible d'être contractée dans le milieu de travail.

Les erreurs typographiques sont très fréquentes, souvent les valeurs nominales sont mal épelées produisant une nouvelle valeur possible pour l'attribut concerné. Différentes appellations peuvent exister pour désigner une même chose. Un fermier, une fermière, un agriculteur ne représentent ils pas un même et unique métier? La liste des valeurs d'une variable nominale doit être examinée avec minutie. Le recensement des différentes valeurs que peut prendre une variable est recommandé quand cela est possible. Les valeurs suspectes doivent être vérifiées et les synonymes, acronymes et homonymes identifiés et traités convenablement.

Pour les valeurs numériques, la plupart des erreurs peuvent être détectées et corrigées facilement. Une visualisation graphique suffirait à détecter les valeurs excentriques ou la répétition anormale de valeurs souvent révélatrices d'erreurs. Mais il arrive que les erreurs ne soient pas décelables, la participation des connaisseurs du domaine d'étude est alors indispensable.

L'erreur délibérée est une autre forme de valeur erronée. L'utilisation d'un faux code postal dans l'urgence d'un hôpital pour recevoir un malade pour lequel on a pas l'adresse exacte et qui n'est pas disposé, vu son état, à la fournir, l'utilisation d'une fausse épellation de son nom ou d'une fausse identité, sont des exemples d'erreurs délibérées qu'il est difficile de détecter. La maîtrise de la sémantique des données est le seul outil capable de déceler de telles erreurs.

Les erreurs dites de mise à jour sont un type très répondu de valeur erronée. La mise à jour des données n'est pas toujours systématique. L'adresse d'un client peut rester la même pendant des années alors qu'il a changé de domicile à plusieurs reprises. Avant d'utiliser des données dans un processus d'extraction de

la connaissance, il est utile de vérifier si les données sont toujours valides et n'ont pas changées depuis la première saisie.

6. LA SÉLECTION DES ATTRIBUTS

La sélection des attributs est une opération nécessaire et utile avant l'utilisation des algorithmes de datamining. Elle est nécessaire pour des raisons pratiques inhérentes aux performances des algorithmes de datamining et aussi pour supprimer les attributs redondants ou non pertinents pour l'objectif établi et dont la présence ne fait qu'altérer la qualité des résultats. La sélection est utile car elle permet de réduire la complexité de la recherche dans l'espace des hypothèses. Plus le nombre d'attributs est petit moins d'hypothèses sur les données peuvent être faites. Cela a l'avantage de faciliter la tâche aux algorithmes de datamining, de réduire les dimensions des données, de supprimer le bruit, de rehausser la compréhension des données et de simplifier l'évaluation des modèles. La sélection d'attributs se fait en fonction de certains critères de mesure. La détermination des critères dépend des objectifs à atteindre.

La sélection d'attributs consiste en la sélection à partir de l'ensemble total d'attributs disponibles dans la base de données d'un sous-groupe d'attributs dans le but de ne soumettre à l'algorithme de datamining qu'un sous-ensemble minimal et suffisant. Il est nécessaire que les attributs choisis satisfassent pleinement l'objectif visé. Une des raisons de la sélection des attributs est de maximiser les performances des algorithmes et la précision des connaissances à découvrir. L'apparition d'attributs non pertinents peut sensiblement augmenter le temps de calcul des algorithmes de datamining parce que la plupart d'entre eux ont une complexité algorithmique en temps d'exécution supérieure à la complexité linéaire [Berkin02].

Il est préférable que l'utilisateur fasse lui même la sélection des attributs pertinents. Mais cela peut s'avérer difficile s'il n'a ni la compétence nécessaire ni l'aide des connaisseurs du domaine pour le faire. Il existe des outils de filtrage, basés souvent sur des théories statistiques, pour aider l'utilisateur à choisir les attributs pertinents. L'utilisation des algorithmes de datamining peut être une autre solution, une première passe d'un algorithme de forage de données permettant de déterminer les attributs significatifs [Witten00]. Les arbres de décision et les règles associatives sont très indiqués pour ce genre de prétraitement. Les techniques de visualisation graphique des données permettent une sélection rapide et intuitive des données [Barioni02], [Razente04]. Il existe deux approches majeures de sélection d'attributs. L'approche par encapsulation et l'approche par filtre.

6.1. Approche par encapsulation (Wrapper)

C'est une méthode de recherche itérative et heuristique. Elle est basée sur une recherche empirique ayant recours aux essais et erreurs pour la résolution du problème. Les données sont séparées en deux ensembles. L'ensemble d'entraînement et l'ensemble d'évaluation. Chaque itération est composée de trois étapes. Premièrement, un sous-ensemble d'attributs appelés attributs candidats est sélectionné en tenant compte de certains critères et en prenant en considération les attributs candidats de l'itération précédente. Deuxièmement, l'algorithme de forage de données est appliqué aux données d'entraînement en ne considérant que l'ensemble des attributs candidats. Troisièmement, la précision des règles ou

modèle découvert est évaluée en utilisant le sous-ensemble d'évaluation. La précision estimée est directement utilisée comme mesure de la qualité de l'ensemble des attributs candidats. L'algorithme de sélection des attributs est utilisé comme une enveloppe autour de l'algorithme de datamining.

La sélection des attributs est souvent une procédure gourmande en ressources machine. Deux méthodes de base de sélection d'attributs existent. La sélection avant (Forward Selection) et l'élimination arrière (Backward Elimination) [Caetano00], [Blum97]. La sélection avant commence avec un ensemble vide d'attributs et sélectionne un attribut à la fois de façon itérative jusqu'à ce qu'aucune amélioration en terme de précision de la classification ne soit possible. L'élimination arrière commence par tout l'ensemble d'attributs et élimine un attribut à la fois jusqu'à ce qu'aucune amélioration en terme de précision de la classification n'est possible. L'élimination arrière a l'avantage de mieux prendre en compte les interactions entre les attributs, ce qui la rend plus adaptée pour les bases de données dont un grand pourcentage d'attributs est pertinent pour l'étude. Alors que la sélection avant est plus efficace si le nombre d'attributs pertinents est petit par rapport au nombre total d'attributs. Il existe d'autres approches de sélection d'attributs plus complexes, mais beaucoup plus lourdes à mettre en œuvre car elles sont de grandes consommatrices de temps d'exécution [Blum97].

6.2. Approche par filtre

Dans cette approche, l'algorithme de sélection d'attributs est totalement indépendant de l'algorithme de datamining. Un même ensemble d'attributs sélectionnés par le filtre peut être utilisé pour différents algorithmes de datamining. Un des filtres les plus connu est le filtre Relief de Konoenko [Fayyad96] qui utilise une méthode aléatoire inspirée de l'apprentissage par cas ou des k plus proches voisins. L'avantage d'un tel filtre est qu'il est assez efficace puisque son temps d'exécution est proportionnellement linéaire au nombre d'attributs et au nombre de sujets utilisés dans l'étude. Un autre filtre proposé pour la première fois par Koller et Sahami [Koller96] se base sur la différence entre la distribution des probabilités des classes ayant comme attributs ceux choisis dans l'ensemble des candidats et la distribution correspondante après l'élimination d'un attribut. La méthode implémente une recherche itérative arrière qui élimine à chaque itération l'attribut dont la suppression minimise la différence entre les deux ensembles attributs. Même s'il est plus robuste que le premier cité, il est utile de préciser que sa complexité algorithmique qui est proportionnelle au nombre d'attributs est quadratique.

L'approche par encapsulation est souvent supérieure à l'approche par filtre. Ceci est assez compréhensible puisque l'approche par encapsulation minimise l'erreur relative en effectuant la sélection des attributs en étroite collaboration avec l'algorithme de datamining. Mais cela se paye par un plus grand temps de calcul. Le compromis entre le taux d'erreur des résultats et les ressources temps d'exécution et mémoire vive nécessaire est souvent rencontré en datamining. Le choix se fait en collaboration avec l'utilisateur final et la nature du problème à résoudre. Une autre raison qui plaide pour l'approche par filtre est qu'il est souvent difficile de déterminer lequel des algorithmes de datamining est le plus adéquat pour le problème à résoudre. Il est donc, préférable de faire une seule sélection d'attributs puis d'essayer les

différentes méthodes dont on dispose pour résoudre le problème étudié que d'utiliser l'approche par encapsulation laquelle fait une sélection d'attributs pour chaque algorithme de datamining qui est lui même appelé plusieurs fois pour chaque sélection. Pour les grandes bases de données le problème ne se pose pas puisque l'approche par encapsulation est tout simplement inapplicable. L'approche par filtre permet d'essayer plusieurs algorithmes ce qui peut accroître sensiblement les connaissances extraites des données.

7. DISCRÉTISATION

Méthode qui consiste à remplacer chaque donnée par un élément d'un ensemble discret préalablement défini. Dans le cas des valeurs continues (entiers, réels), la détermination de l'ensemble des valeurs discrètes est obtenue par le partage de l'intervalle des valeurs continues en une liste finie d'intervalles. Cela permet de convertir un attribut de type continu en un attribut de type catégorique où chaque intervalle est considéré comme une valeur discrète de l'attribut.

Les algorithmes de discrétisation sont de deux types, les algorithmes dits aveugles et les algorithmes dirigés. Les premiers discrétisent un attribut sans prendre en considération les valeurs de la classe visée. Par exemple, l'égalité des intervalles peut être le critère de discrétisation qui donnera en sortie un nombre prédéfini d'intervalles de longueurs égales.

Les algorithmes dirigés ne se contentent pas de considérer l'attribut à discrétiser mais prennent en compte les valeurs de la classe visée lors de la discrétisation. Il est évident que l'algorithme dirigé est plus performant que celui de type aveugle quand les intervalles proposés sont directement impliqués dans le processus de classification. Pour illustrer cela, supposons qu'on a un test médical dont les valeurs varient entre 0 et 200. Un algorithme de type aveugle produirait des intervalles comme 0-20, 21-40,..., 181-200. Si nous supposons que le seuil où le test passe du positif au négatif est la valeur 30, aucun algorithme de datamining ne pourra produire de bons résultats avec une telle discrétisation. Alors qu'un algorithme dirigé produira probablement deux intervalles séparés par la valeur 30.

La plus simple des discrétisations est d'essayer de déterminer pour chaque classe un intervalle correspondant qui regroupera le maximum possible d'objets de cette classe. Des recherches itératives et heuristiques sont utilisées. Elles reposent sur la comparaison de diverses valeurs des limites supérieures et inférieures d'un intervalle donné. Les critères de comparaison sont des mesures qualitatives qui peuvent dépendre des données elles mêmes. Le test statistique Khi-2 est souvent utilisé [Tay02]. Les fonctions de gain d'information sont aussi utilisées [Dougherty95].

Le premier avantage de la discrétisation est qu'elle permet d'étendre l'exploration des données aux algorithmes qui ne supportent pas les valeurs continues. Aussi, la discrétisation peut améliorer sensiblement l'interprétation des connaissances en sortie. Un troisième avantage de la discrétisation est la réduction du temps de calcul des algorithmes de datamining. Le temps de traitement des valeurs continues étant sensiblement plus grand que celui des valeurs discrètes. Des études rapportent que le

temps de calcul peut être divisé par 10 voir 30 ou même 50 pour les bases de données à plusieurs attributs continus et cela sans grande perte dans la précision de la classification [Pappa04], [Freitas96].

L'inconvénient de la discrétisation est qu'elle peut causer un plus grand taux d'erreur puisqu'elle inhibe certains détails de l'attribut qu'elle traite qui peuvent être cruciaux. Cet inconvénient dépend des données traitées, de l'objectif à atteindre et de la méthode de datamining utilisée. Pour les grandes bases de données, il est recommandé de procéder à une discrétisation des données continues dans la mesure du possible, sans pour autant, tomber dans la facilité et utiliser les algorithmes de type aveugle lesquels restent déconseillés dans la plupart des cas.

8. L'ÉCHANTILLONNAGE

L'échantillonnage est un puissant moyen qui permet d'estimer des caractéristiques d'une population donnée à un moindre effort, avec fiabilité et sans biais, en étudiant un sous ensemble représentatif de la population. La précision de l'estimateur de la taille de l'échantillon dépend du degré de variabilité de la population. Néanmoins, il existe deux facteurs à considérer pour accroître la précision d'un estimateur. Le premier est l'augmentation de la taille de l'échantillon car il est démontré que la variabilité de l'estimation est inversement proportionnelle à la racine carrée de la taille de l'échantillon. Le deuxième est le choix de la bonne méthode d'échantillonnage.

8.1. Techniques de l'échantillonnage

L'Échantillonnage aléatoire simple

Chaque individu de la population a la même probabilité d'être sélectionné dans l'échantillon. Le résultat est un échantillon dont les caractéristiques des individus sont distribuées de façon similaire sur l'ensemble de la population. Le principe est d'allouer à chaque individu un numéro correspondant à sa position dans l'ensemble de population. Un nombre aléatoire est généré et l'individu dont le numéro correspond au nombre généré est sélectionné dans l'échantillon. Cela peut générer un échantillon atypique. L'échantillonnage est de deux types, avec et sans remise. Dans le premier type un individu peut être sélectionné plus d'une fois dans l'échantillon alors que dans le deuxième type un individu ne peut être choisis qu'une seule fois. Les deux types sont sensiblement équivalents si la population d'origine est grande parce que la probabilité de choisir un individu plus d'une fois est quasiment nulle.

L'Échantillonnage aléatoire stratifié

Ce type d'échantillonnage commence d'abord par diviser la population en strates ou groupes ayant des caractéristiques communes et d'opérer un échantillonnage aléatoire simple sur chaque strate. Cela procure deux avantages, le premier est le contrôle du nombre d'individus dans l'échantillon appartenant à chaque strate, ce qui permet de privilégier une strate en augmentant le nombre d'individus lui appartenant. Une strate sous-représentée dans une population peut être par conséquent mieux étudiée et influera plus le modèle résultant. Le second avantage est qu'il sera possible d'avoir un plus petit taux d'erreur à l'intérieur de chaque strate plutôt qu'un taux d'erreur global relativement grand du à la variabilité

entre les individus de strates différents. L'échantillonnage aléatoire stratifié est à utiliser avec précaution car il peut rehausser la présence d'individus particuliers et causera une surreprésentation de ceux-ci.

L'Échantillonnage par groupage

Il arrive que les individus d'une population forment des groupes naturellement distincts. Cette caractéristique est récupérée pour faire de l'échantillonnage. Deux approches sont appliquées, la première est d'échantillonner chaque groupe individuellement ce qui revient à l'échantillonnage aléatoire stratifié. La deuxième approche est d'opérer un premier échantillonnage sur les groupes eux même, ce qui signifie choisir aléatoirement un certain nombre de groupes et de prendre les individus des groupes ainsi sélectionnés. L'échantillonnage à l'intérieur des groupes sélectionnés peut aussi être envisagé, on parlera alors l'échantillonnage à deux niveaux.

Si on peut établir que l'ensemble de la population à étudier est constitué de groupes distincts, il est fort probable que les individus d'un même groupe soient plus similaires entre eux qu'avec un individu d'un autre groupe. Ce qui contredit les suppositions de départ de la plupart des approches statistiques d'échantillonnage lesquels supposent la non dépendance des données et retourneront, forcément dans ce cas, des résultats avec biais.

Pour les grandes bases de données, il arrive qu'on procède à la sélection aléatoire de blocs de données, pour ensuite utiliser toutes les données contenues dans les blocs. Le choix d'une telle approche est dû à des considérations techniques et de performance. Pour lire un enregistrement d'un bloc de données, il faut charger en mémoire l'intégralité du bloc, la lenteur des entrées-sorties et le grand volume des données à explorer plaide pour une telle approche d'échantillonnage [Chaudhuri98].

L'Échantillonnage systématique

Si les individus d'une population sont numérotés, il est parfois avantageux d'utiliser l'échantillonnage systématique. L'ensemble de la population est divisé en intervalles de longueur k. Un individu est choisi aléatoirement sur l'intervalle 1 à k, après ça, chaque k-ème individus à partir de la position courante est sélectionné. Ce type de sélection n'est pas aléatoire, l'échantillon peut ne pas être représentatif de toute la population mais il peut être utile dans certains cas particuliers.

L'Échantillonnage à deux phases

L'échantillonnage à deux phases est utile si on veut organiser notre échantillon en fonction des valeurs d'une ou de plusieurs variables, et on ignore la distribution des variables dans la population. On optera alors pour un premier échantillonnage qui nous renseignera mieux sur les données pour ensuite faire un deuxième échantillonnage plus éclairé. Un échantillonnage à deux phases peut aussi être utilisé pour l'estimation de la taille du deuxième échantillon en fonction de certains paramètres qu'un échantillonnage initial révélera.

8.2. Taille de l'échantillon

Il existe des méthodes statistiques simples qui permettent d'estimer la taille minimale de l'échantillon pour une bonne représentation de la population. Ces outils sont basés sur l'estimation des moyennes et les variances des variables de la population à être échantillonnée. Si les estimateurs sont difficiles à calculer, un échantillonnage à deux phases peut être utilisé pour les générer. Dans le cas de l'échantillonnage par groupage ou l'échantillonnage aléatoire stratifié, la taille de l'échantillon peut être estimée pour chaque groupe. La précision d'estimation de l'échantillon peut différer d'un groupe à l'autre. La taille de l'échantillon global est la somme des tailles de tous les groupes. Une bonne approche d'estimation de la taille de l'échantillon est de continuer à ajouter un nouvel individu à l'échantillon jusqu'à ce que les estimateurs telles que la moyenne et la variance ne changent plus ou peu, c'est ce qu'on appelle l'échantillonnage progressif ou adaptatif [Provost99].

9. ESTIMATION DES ERREURS ET EVALUATION DES CONNAISSANCES

Les algorithmes de datamining utilisent les estimations d'erreurs pour faire des choix cruciaux qui détermineront le modèle en sortie. Le concepteur d'un modèle évalue les algorithmes en estimant les erreurs des modèles produits par l'algorithme.

L'erreur produite par un modèle peut être calculée en comparant les valeurs estimées aux valeurs réelles connues. L'erreur peut être estimée par divers moyens tels le pourcentage des instances mal classées, le gain, la fonction khi-carré et autres. Les tables de contingences ou matrices de confusion sont simples et faciles à comprendre, elles résument bien les résultats et permettent de facilement calculer les taux d'erreurs globaux et spécifiques à chaque classe ou type de valeurs estimées. La fonction khi-carré est un bon résumé des tables de contingences.

La variance et le biais sont deux critères souvent utilisés pour décrire la qualité des scores produits par un estimateur. Le biais d'un estimateur est la différence entre la valeur de l'espérance des scores $E(X)$ et la valeur du paramètre sur la population totaleΨ. Un estimateur est considéré sans biais si $E(X)=\Psi$. Alors que la variance mesure la dispersion des scores autour de la moyenne. Dans le choix entre deux estimateurs sans biais, celui dont la variance est la plus petite est le meilleur. Il arrive même qu'un estimateur biaisé soit préférable à un estimateur sans biais, si la variance du premier est beaucoup plus petite que celle du deuxième.

L'estimation de l'erreur peut aussi se faire par la fonction de *Perte*. Cette dernière mesure, par exemple, le coût généré par les individus mal classés. La fonction est incrémentée de un si un individu est mal classé et de zéro s'il est bien classé. Pour les valeurs continues, la perte peut être estimée par la perte quadratique, soit la différence entre les carrées des valeurs estimées et des valeurs réelles. La perte quadratique est préférable dans la plupart des cas. Elle peut par exemple donner différentes valeurs de perte pour différents types de mal classement.

Les tests par hypothèse sont une autre forme d'interprétation des scores. L'idée est de faire une déduction statistique sur une population à partir du score d'un échantillon. Le test d'hypothèse compare le score de l'échantillon à la distribution des scores. L'hypothèse nulle H_0, de départ, est considérée comme vraie. Une fois la distribution d'échantillonnage calculée, un test d'hypothèse est posé pour estimer la probabilité qu'un score donné sera atteint, ce qui vérifiera que l'hypothèse nulle, H_0, est vraie. La probabilité estimée est connue sous le non de la *P-value*. Pour accepter ou rejeter H_0 la *P-value* est comparée à α, un seuil de probabilité prédéterminé qui indique la probabilité maximale acceptable pour le rejet de H_0. Si P est inférieure ou égale à α, alors l'hypothèse H_0 est rejetée sinon elle est acceptée. Le choix de α est à faire avec précaution avec l'aide d'un statisticien si possible. Néanmoins, des valeurs comme 0.05, 0.01 et 0.001 sont souvent utilisées.

Il arrive qu'on ait à faire des choix entre plusieurs items disponibles, où un item peut représenter un modèle, un composant ou un paramètre. Les comparaisons multiples sont des tests de choix utilisés dans ce but. Elle s'exécute en trois étapes. Premièrement, différents items sont générés, puis un score est estimé pour chaque item en utilisant l'échantillon d'apprentissage, et finalement l'item qui a le meilleur score est sélectionné. Le choix d'un attribut dans la génération d'un arbre de décision, le choix d'un modèle dans la construction d'un méta-modèle sont des exemples de choix par comparaisons multiples. La validation croisée est une méthode qui peut être associée à la comparaison multiple pour le choix des items. La validation croisée divise l'échantillon E d'apprentissage en k sous-ensembles disjoints de tailles $1/k$. Un algorithme est appliqué k fois aux échantillons E-E_i, où E_i représente le i-ème sous-ensemble de l'échantillon E. les résultats de l'algorithme sont alors évaluées sur le sous-ensemble E_i produisant ainsi k différents scores. Les k scores sont alors combinés pour produire un seul score agrégé.

10. TECHNIQUES DU DATA MINING

Les techniques de datamining sont une réponse des informaticiens à l'incapacité des techniques de statistique et d'analyse de données à répondre à des demandes urgentes des décideurs. Ces derniers avaient besoin d'outils d'aide à la décision qui ne soient pas seulement des tableaux de chiffres difficiles à comprendre et à interpréter. Les techniques de datamining offrent une autre manière d'analyser et de traiter les problèmes et une certaine flexibilité à les modéliser sans trop se soucier de la rigueur mathématique, condition nécessaire pour pouvoir modéliser la complexité du monde réel.

On peut regrouper les techniques de datamining majeures dans quatre catégories distinctes. La première catégorie regroupe les analyses statistiques, telles que la régression linéaire, la régression logistique et le test du Khi2. La deuxième catégorie regroupe les techniques de classification et de segmentation qui sont des techniques dites supervisées puisqu'elles construisent des modèles en fonction de classes connues. Des variantes de ces techniques permettent aussi de faire l'estimation des valeurs d'un attribut ou carrément la prédiction. La troisième catégorie regroupe les techniques de groupage de données, appelées aussi techniques de classification non supervisée. Elles ont pour but de regrouper les données similaires dans des classes distinctes non connues. Ces techniques sont moins étudiées que ceux de la

classification supervisée, probablement, parce que plus difficiles à valider. L'interprétation des résultats issus de ces techniques reste de la compétence exclusive des connaisseurs du domaine lesquels doivent être partie intégrante de l'équipe de construction de tels modèles. La quatrième catégorie est l'extraction des règles associatives qui est la seule technique issue de la recherche dans la discipline des bases de données et permet de mettre en évidence les dépendances entre les attributs en extrayant des règles les associant. En plus de ces quatre grandes catégories, on peut citer les analyses multi-niveaux d'agrégation et de synthèse des données telles que les techniques OLAP utilisées dans les entrepôts de données. Les techniques de calcul des similarités entre objets utilisées par exemple dans la recherche d'objets complexes tels la recherche de la photo d'une personne à partir d'un prototype ou de critères définis par l'utilisateur. Le calcul des similarités peut être utilisé soit pour ressortir tous les objets répondants à des critères de départ ou pour regrouper les objets considérés comme similaire en fonction de ces critères [Li99, Yan05]. Des techniques de modélisation de la navigation d'un utilisateur à travers des pages WEB commencent aussi à émerger. Elles essayent de comprendre, modéliser puis interpréter les comportements des utilisateurs. Elles ont pour but la remise en cause des designs des sites WEB pour une plus grande accessibilité de l'utilisateur mais aussi, elles essayent de capturer les informations nécessaires pour un meilleur marketing.

10.1. Analyse statistique

Dans les méthodes statistiques, on cherche à explorer les corrélations entre variables, pour cela il faut trouver une fonction qui détermine un ensemble de variables. Nous allons décrire les deux méthodes statistiques simples que sont la régression linéaire et la régression logistique.

Régression linéaire :

Technique consistant à approcher la dépendance entre deux variables X et Y par une droite

$Y = a X + b$

Dans le cas le plus simple, cette technique tente d'approcher par une droite la relation entre deux variables. On peut l'appliquer à plusieurs variables pour essayer de déterminer une droite qui passe le plus proche possible de la courbe définie par les dites variables. L'utilisation d'une telle approche n'est pas évidente dans la mesure où rare sont les phénomènes qui sont linéaires. L'exemple suivant montre la dépendance entre le taux de la troponine dans le sang humain et le risque d'avoir un infarctus du myocarde. Les points de la *figure 03* ne vérifient presque jamais la droite. Dans le cas de phénomènes linéaires, la régression linéaire reste simple et efficace.

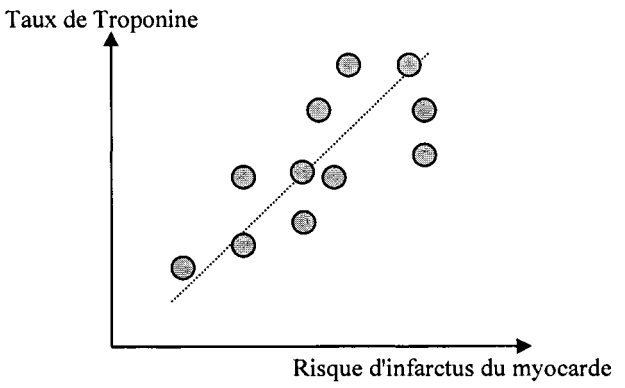

Figure 03 : Exemple de régression linéaire

Régression logistique :

Permet de prendre en compte le cas des variables binaires ("0,1", "vrai, faux"). C'est une extension de la régression linaire au cas discret. Comme il n'est pas possible de représenter une variable binaire par une droite, on calcule la dérivée seconde $Y'' = log\ (p/(1-p))$ qui représente le logarithme du rapport du nombre de chances de succès sur le nombre de chances d'échec. Comme Y'' est un réel, le problème est ramené à l'étude des dépendances entre deux variables de type réel. Une fois Y'' déterminé, il est facile de calculer Y en déduisant la probabilité p. La régression logistique est une régression linéaire du type $Y'' = a\,X + b$.

10.2. Les réseaux bayésiens :

La technique est basée sur le théorème de Bayes qui détermine la probabilité conditionnelle selon la formule suivante :

$$P(Ai/B) = P(Ai)*P(B/Ai) / \Sigma_j P(Aj)*P(B/Aj)$$

Intuitivement, la formule permet de prédire le future sachant le passé, en supposant la reproductibilité des probabilités. Si les événements A_i sont indépendants et ayant observé B, elle permet d'estimer les probabilités des événements A_i futurs, en fonction des événements A_i passés ayant eu B pour conséquence.

Un petit exemple de la gestion d'infirmiers permet d'illustrer l'utilisation des réseaux bayésiens. Les appels des patients peuvent venir de la chambre une, deux ou trois. Les patients peuvent sonner l'infirmier en utilisant deux boutons, l'un rouge pour les urgences et un autre vert pour les cas courants. Chaque fois

qu'un patient sonne un voyant s'allume, il est de couleur rouge ou verte. Grâce aux données de la base, on peut calculer la probabilité de présence d'une couleur en fonction de l'avènement d'un appel d'une des trois chambres. On calcule les probabilités a priori des appels provenant de chacune des chambres et celles d'avoir un voyant vert ou rouge *(tableaux 01)*.

En utilisant le théorème de Bayes, on peut déterminer la chambre d'appel en fonction de la couleur observée et la probabilité d'avoir un appel d'une chambre donnée.

Si on suppose que le service à l'étage est assuré par deux infirmiers A et B, et que les tâches leurs sont réparties comme indiqué dans le *tableau 02* ci-dessous, on peut construire le réseau bayésien modélisant les données de notre base. Réseau à partir duquel, il est possible de calculer la probabilité d'apparition d'un infirmier selon l'avènement d'une couleur ou de l'autre. Les *tableaux 01*, *02* et *03* et la *figure 04* suivants résument la construction d'un réseau bayésien.

P(Couleur/Chambre)	Rouge	Vert	P(Sonnerie)
Chambre 01	0.70	0.30	0.20
Chambre 02	0.40	0.60	0.10
Chambre 03	0.20	0.80	0.70
P(Couleur)	0.32	0.68	1.00

Tableau 01: Probabilités conditionnelles et a priori

P(Couleur/Sonnerie)	Rouge	Vert
Chambre 01	0.44	0.09
Chambre 02	0.12	0.09
Chambre 03	0.44	0.82

Tableau 02 : Probabilités a posteriori calculés à l'aide du théorème de Bayes

P(Infirmier/Chambre)	Chambre 01	Chambre 02	Chambre 03
Infirmier A	0.70	0.40	0.50
Infirmier B	0.30	0.60	0.50

Tableau 03 : Probabilité conditionnelle Infirmier/Chambre

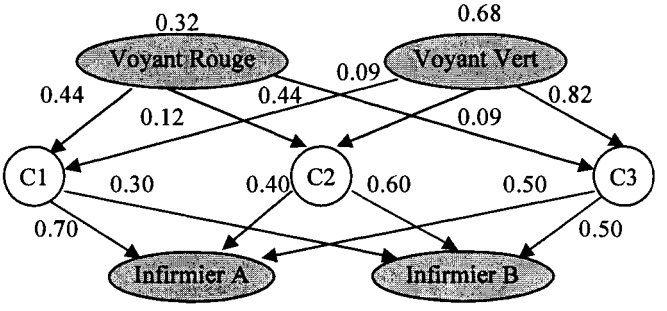

Figure 04 : Exemple de réseau bayésien

10.3. Réseaux de neurones

Un réseau de neurones est, comme son nom l'indique, constitué de composants élémentaires appelés neurones. Modélisant un neurone biologique, le composant de base du réseau est une cellule à *n* entrées *E1, E2...* En, et une sortie. Chaque entrée *Ei* possède un poids *Wi*. Le neurone combine les *n* entrées sous la forme d'une fonction linéaire *ΣWi* Ei*, puis applique une fonction de transfert *f* au résultat afin d'obtenir la sortie. La fonction *f* est, généralement, une fonction de seuil, qui change complètement la sortie, si une petite modification est appliquée au entrées.

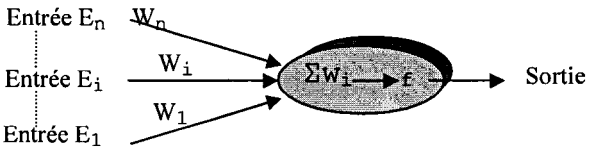

Figure 05 : Représentation d'un neurone

Un réseau se compose de multiples neurones interconnectés. Il est en général, organisé en couche, où chaque neurone de la couche *i* recevant en entrée certaines sorties des neurones de la couche *i-1*. Trois couches sont généralement suffisantes, l'une servant au codage, l'autre à la modélisation de l'intelligence et la troisième à la préparation de la sortie.

Lors de l'apprentissage, le réseau commence par chercher un modèle en analysant un échantillon de données. Pour cela il injecte des entrées à sorties connues, les propage à travers le réseau, calcule la différence entre la sortie souhaitée et celle obtenue, puis fait une propagation arrière de cette même différence afin de corriger les poids en entrées en augmentant les poids des neurones qui donnent de bons résultats et en diminuant ceux des neurones qui donnent de mauvais résultats. Il existe plusieurs types de réseaux de neurones, les uns spécialisés en classification et prédiction et d'autres en groupage des données. Leurs apprentissages sont toujours difficiles à faire mais les résultats qu'ils produisent sont généralement supérieurs aux autres modèles de datamining.

10.4. Groupage de données

Ces techniques se basent sur la recherche des similarités et des différences dans une population d'individus de même type afin d'identifier les individus similaires et les mettre dans un même groupe. La mesure des similarités dans les techniques de groupage est réalisée grâce à une fonction de distance. Il est donc indispensable de définir cette dernière pour pouvoir faire une bonne classification des individus par groupage. Une fonction de distance *d* obéit aux règles suivantes :

Quelque soit les individus *A* et *B* de l'ensemble de population :

- $d(A,B) > 0$
- $d(A,A) = 0$
- $d(A,B) = d(B,A)$
- $d(A,B) \leq d(A,C) + d(C,B)$

10.4.1. Méthode des K-Means (Centres Mobiles) :

Elle permet de classer les individus en *N* groupes en les agrégeant autour de centres mobiles *(figure06)*. Elle s'exécute en quatre étapes essentielles :

1. *Choisir K individus initiaux, appelés centroides.*
2. *Placer les autres individus dans le groupe de centre le plus proche en utilisant la fonction de distance*
3. *Recalculer un centre fictif pour chaque groupe créé*
4. *Itérer l'algorithme jusqu'à ce que les individus ne changent plus de groupes*

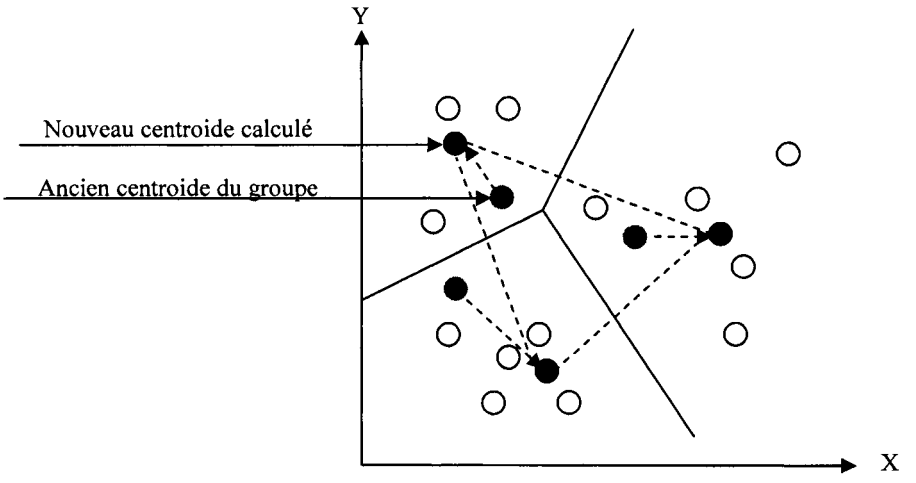

Figure 06 : Illustration des K-means

Le choix des k-centroides initiaux est fait de façon aléatoire. Mais pour des raisons d'efficacité et de performance d'autres critères peuvent être utilisés pour le meilleurs choix possible des centroides. Cela facilitera la convergence de l'algorithme en minimisant le nombre d'itérations et évitera de converger vers un centroide local. Les itérations remettent en question les choix précédents et les corrigent.

10.4.2. Méthode par agglomération (Groupage hiérarchique)

L'idée est de regrouper deux à deux les individus les plus proches de sorte à former de nouveaux individus que l'on regroupe à leur tour. Pour déterminer les individus les plus proches, on construit la matrice des distances entre les *n* individus. À la fin de la première itération un ensemble de *(n/2)+1* ou *n/2* de groupes est formé. Lors de l'itération suivante chaque groupe est considéré comme un simple individu représenté par son centre. La *figure 07* illustre un exemple d'implémentation du groupage hiérarchique par agglomération. L'ensemble de la population est représenté par les individus {2, 5, 8, 12, 17, 22}.

Le choix de la fonction de distance est très important et influe directement sur la construction des groupes. Dans le cas ou un individu est à la même distance de deux autres, le choix de le grouper avec l'un ou l'autre peut considérablement modifier la configuration des groupes finaux parce qu'un tel choix va se répercuter en cascade sur les regroupements futures. Des variantes ont été proposées pour remettre en cause les choix de regroupement après chaque itération [Gardarinii99].

Distance	2	5	8	12	17	22
2	0	**3**	6	10	15	20
5	3	0	**3**	7	12	17
8	6	3	0	**4**	9	14
12	10	7	4	0	**5**	10
17	15	12	9	5	0	**5**
22	20	17	14	10	5	0

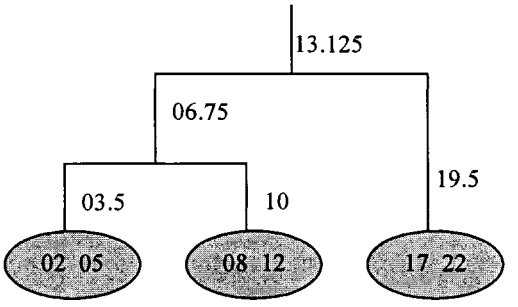

Figure 07 : Illustration du Groupage par Agglomération

10.4.3. Méthode par voisinage dense

Technique qui repose sur deux concepts, le voisinage d'un individu et la distance entre deux individus. Le voisinage d'un individu est déterminé par la fonction de distance *d* et un seuil minimal ε au-dessus duquel on sort du voisinage. La densité du voisinage d'un individu est le nombre de voisins qui l'entourent. Deux points sont dans un même voisinage si la distance qui les sépare est inférieure au seuil ε. Un voisinage est dit dense s'il contient plus qu'un minimum *k* d'individus.

- *Voisin (A,B)* $\Leftrightarrow$ *distance (A,B)* $\leq \varepsilon$
- *Dense (A)* $\Leftrightarrow$ *nombre (voisins(A))* $\geq k$

Pour déterminer les groupes, la méthode commence par choisir aléatoirement un point dense. Tous les points qui sont atteignables à partir de ce point, selon le seuil de densité établi, forment un groupe. Les points atteignables sont les voisins directes, les voisins des voisins et ainsi de suite.

Le choix de la fonction de distance *d*, du seuil ε et du nombre *k* de voisins qui déterminera la densité d'un voisinage sont des choix qui détermineront le nombre de groupes et leurs compositions. Bien que cet algorithme soit simple à implémenter et donne relativement de bon résultats, il est peu utilisé à cause des temps de calcul et son inadaptation au grandes bases de données.

10.5. Segmentation des données

Technique de classification souvent basée sur les arbres de décision pour ranger les individus en classe. Un arbre de décision est un arbre permettant de classer les individus en sous-classes par divisions hiérarchiques, dans lequel un nœud représente une sous-classe du nœud parent et un arc représente un prédicat de placement des individus de la classe parente dans la sous classe.

Un arbre de décision peut être interprété comme un ensemble de règles de déduction :

Si Client = «Client1» et état facture = «payé» alors solvable;

La *figure 08* ci-dessous, illustre un arbre de décision d'une table relationnelle décrivant l'état de facture des clients ainsi que leur solvabilité. L'arbre peut être utilisé pour prédire, en fonction de l'état d'une facture d'un client s'il est solvable ou pas. On remarque aussi que l'arbre n'est qu'une représentation hiérarchique de la table relationnelle.

Client	Etat facture	Solvabilité
Client1	payée	solvable
Client1	payée	solvable
Client2	payée	solvable
Client2	en attente	insolvable
Client3	impayé	solvable

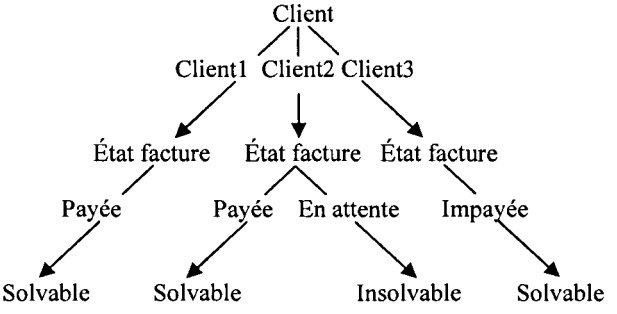

Figure 08 : Exemple d'arbre de décision

Un arbre qui reflète fidèlement une table relationnelle, sans perte d'information est dit arbre complet. Il est extrêmement volumineux, et de peu d'utilité. Pour avoir un petit arbre simple et utile pour la prise de décision et éviter ainsi la sur-spécialisation de l'arbre qui rendrait ce dernier très performant sur les données d'entraînement et éventuellement sur les données du tests mais complètement inefficace sur de

nouvelles données, les algorithmes de construction d'arbres de décision utilisent des techniques d'élagage pour alléger l'arbre de certaines de ses branches. Le choix des branches à élaguer peut se faire selon le niveau de profondeur de la branche, du nombre de cas constituant la branche, en déterminant l'utilité d'une division par le calcul de son coût, ou sur d'autres critères. La construction de l'arbre se fait en deux étapes :

1. *Construction d'un arbre par division récursive des nœuds : à chaque étape, il faut choisir le meilleur attribut sur lequel peut s'effectuer la division*
2. *Élagage de l'arbre récursivement depuis les feuilles dans le but de réduire sa taille.*

10.6. La découverte des règles associatives

Une règle associative est une règle du type $X \rightarrow Y$ où X et Y sont des éléments d'individus, traduisant le fait que si les individus X son présents dans une transaction, alors les individus Y le sont avec une certaine probabilité. Le terme transaction est hérité de la vente de détail et est utilisé dans les règles associatives plutôt que les termes enregistrement ou instance. Une deuxième raison qui légitimerait son utilisation est qu'une transaction n'a pas de taille fixe. Elle peut contenir un ou des centaines d'articles alors qu'un enregistrement est de longueur prédéterminée et fixe.

Par des règles du type $X \rightarrow Y$, on cherche à associer deux ou plusieurs produits selon la régularité de leur cooccurrence. Comme une règle ne vérifie jamais tous les cas d'une base, il est souvent utile de lui associer une pondération sous la forme d'une probabilité. Pour mesurer le poids d'une règle on utilise deux indicateurs, le premier mesure la fréquence de la règle et le deuxième sa confiance.

Support d'une règle : Mesure du pourcentage de transactions qui vérifient une règle

Support $(X \rightarrow Y) = |X\&Y|/|BD|$

Ou $|X|$ *: le nombre de transactions contenant l'ensemble X*

et $|BD:|$ *le nombre total de transactions*

Confiance d'une règle : Mesure le pourcentage de transactions qui vérifient la partie gauche d'une règle associative parmi celles qui vérifient la partie droite.

Confiance $(X=>Y) = P(Y/X) = |X\&Y| / |X|$

Utilisé tout seul, le support élimine les règles qui sont très valides mais peu fréquentes dans une base de données. C'est pourquoi il faut prendre en compte le deuxième indicateur de confiance. Un exemple simple de règle associative valide et peu fréquente est :

Pain pour Hamburger $\rightarrow$ *Fromage fondu*

On n'achète pas tous les jours du pain pour hamburger, mais si on l'achète, on l'accompagne souvent des aliments qui vont avec. Bien qu'une telle règle ait un faible support, elle reste très valide.

Principe de base de recherche des règles intéressantes

- Un ensemble d'individus est dit fréquent s'il a un support supérieur à un minimum déterminé *minsup.*
- Un k-ensemble fréquent est un ensemble fréquent de taille *k.*
- Tout k-ensemble fréquent est composé de sous-ensembles fréquents.
- Ayant trouvé tous les k-ensembles fréquents, il est possible de générer les (k+1)-ensembles fréquents, en unissant les k-ensembles fréquents deux à deux.
- Continuer la génération des ensembles fréquents jusqu'à ce qu'il n'y ait plus d'ensembles fréquents de cardinalité supérieure.

Pour illustrer ces principes, supposons qu'on a trois individus *A,B* et *C* et des supports minimaux *minsup1, minsup2, minsup3* pour respectivement les ensembles fréquents de tailles un, deux et trois, alors :

- *{A}, {B}, {C}* sont des 1-ensembles fréquents s'ils ont chacun un support supérieur à *minsup1*

Des 1-ensembles fréquents, on peut déduire des 2-ensembles fréquents

- De *{A}* et *{B}* on déduit *{A,B}* un 2-ensemble fréquent si *{A,B}* a un support supérieur à *minsup2*
- De *{A}* et *{C}* on déduit {A,C} un 2-ensemble fréquent si *{A,C}* a un support supérieur à *minsup2*
- *De {B}* et *{C}* on déduit *{B,C}* un 2-ensemble fréquent si *{B,C}* a un support supérieur à *minsup2*

Si {A,B}, {A,C} et {B,C} sont des 2-ensembles fréquents, on peut en déduire les 3-ensembles fréquents

- *De {A,B}* et *{A,C}* on déduit *{A,B,C}* un 3-ensemble fréquent si *{A,B,C}* a un support supérieur à *minsup3*
- *De {A,B}* et *{B,C}* on déduit *{A,B,C}* un 3-ensemble fréquent si *{A,B,C}* a un support supérieur à *minsup3*
- *De {A,C}* et *{B,C}* on déduit *{A,B,C}* un 3-ensemble fréquent si *{A,B,C}* a un support supérieur à *minsup3*

Dans cet exemple, si *{A,B,C}* est fréquent alors *{A}*, *{B}*, *{C}*, *{A,B}*, *{A,C}*, *{B,C}* sont des ensembles fréquents.

10.7. Combinaison de modèles

L'idée de combinaison de modèles reprend le processus naturel de prise de décision. Dans un cas complexe pour lequel il existe plus d'un avis, le décideur a souvent recours à plusieurs experts pour se forger une conviction et prendre une décision. C'est exactement ce que fait la combinaison de modèles en exploitant et évaluant les résultats de plusieurs modèles pour en déduire un résultat. Les méthodes les plus connues de combinaison de modèles sont le bagging, le boosting et le stacking. La combinaison de plusieurs modèles peut augmenter les performances de classification ou de prédiction mais elle a l'inconvénient de rendre le processus de prise de décision plus lourd et plus complexe et parfois difficile à interpréter. Le trois méthodes sus-citées sont des techniques générales qui peuvent être utilisées pour la prédiction numérique et la classification en association avec différents algorithmes de datamining.

10.7.1. Bagging

Le nom bagging est dérivé de l'expression *"Bootstrap aggregating"*. L'idée est d'avoir plusieurs échantillons avec lesquelles sont construits un nombre équivalent de modèles de datamining. Pour chaque instance (enregistrement) des données initiales est calculé sa classe respective en utilisant chacun des modèles. Pour une instance donnée, la classe la plus fréquemment prédite par les différents modèles est choisie comme classe de l'instance. Pour avoir plusieurs échantillons d'entraînement et de test, les données de départ sont transformées en supprimant aléatoirement un nombre d'instances de départ et en dupliquant d'autres pour remplacer ceux supprimées et garder la même taille des données de départ. Les ensembles de données ainsi produits sont utilisés pour l'entraînement et les instances supprimées son utilisées pour les tests. La combinaison de modèles produits est généralement plus performante que l'utilisation d'un seul modèle obtenu en utilisant la totalité de l'échantillon initial.

Pour la prédiction des valeurs numériques, il suffit de prendre la moyenne des valeurs prédites par les différents modèles comme la valeur à prédire de l'instance considérée. Il est aussi possible d'utiliser un vote pondéré ou de calculer une moyenne pondérée au lieu d'un vote et une moyenne simple. Si les données initiales sont stables, l'apport du bagging est insignifiant puisque les modèles générés seront sensiblement les mêmes.

L'algorithme de l'approche bagging est résumé comme suit [Witten00]:

Génération du modèle

Pour chacune des itérations Faire

Créer un nouvel échantillon en faisant des remplacements dans l'échantillon initial

Créer un modèle en appliquant l'algorithme de datamining

Classification

Pour chacun des modèles générés

Prédire la classe de chaque instance

Assigner à chaque instance la classe la plus fréquemment prédite par les modèles

10.7.2. Boosting

L'idée est la même que pour le bagging. Le boosting utilise le vote multiple pour la classification et le calcul de la moyenne pour la prédiction des valeurs numériques en combinant des modèles de même type, générés à partir de différents échantillons. Sauf que le boosting est un processus itératif, où chaque modèle généré est directement influencé par le modèle précédemment produit. Le nouveau modèle favorise les instances mal classées par le modèle précédent. À chaque itération, le boosting donne un poids à chaque instance. L'erreur de classification n'est plus la fraction des instances mal classées sur le nombre total des instances mais la somme des poids des instances mal classées divisé par la somme des poids de toutes les instances. Aussi, le boosting détermine la contribution du modèle en fonction de sa performance plutôt que de considérer les modèles comme égaux. L'algorithme du boosting est comme suit [Witten00]:

Génération du modèle

*Un même poids **p** est attribué à chaque instance*

*Pour chacune des **t** itérations Faire*

Créer un modèle en appliquant l'algorithme de datamining

*Calculer l'erreur **e** du modèle en utilisant les poids des instances*

Si e = 0 ou e ≥ 1/2 : Terminer la génération du modèle

Pour chaque instance des données initiales

*Si l'instance est bien classée par le modèle : p = p * e / (1-e)*

// Normaliser pour que la somme des poids des instances reste la même

*normaliser le poids de toutes les instances : p = p * (∑ anciens p) / (∑ nouveaux p)*

Classification

Un poids de valeur zéro est attribué à toutes les classes

*Pour chacun des **t** (ou moins) modèles Faire*

// pondérer le poids de l'instance pour un modèle en fonction de l'erreur total du

// modèle

Ajouter -log(e / (1 - e)) au poids de chaque classe prédite par le modèle

Retourner la classe avec le plus grand poids

Le boosting peut être adapté aux algorithmes de datamining qui n'acceptent pas les valeurs avec poids. Pour cela, il suffit de générer à chaque nouvelle itération un échantillon à partir du précédent en utilisant les poids des instances. Le même principe que le bagging est utilisé, sauf que les probabilités de choix des instances à supprimer et à dupliquer ne seront plus égales pour toutes les instances mais dépendront du poids de chaque instance.

10.7.3. Stacking

Inventé par David Wolpert, le stacking est un méta-modèle qui combine des modèles de différents types plutôt que des modèles de même type comme dans le cas du bagging et boosting. Combiner plusieurs modèles de types différents en utilisant le vote majoritaire ou en choisissant tout simplement le modèle le plus performant sont les deux méthodes les plus triviales de méta-modèle. Le problème avec le vote est qu'il peut générer de très mauvais résultats si la majorité des modèles ne sont pas performants, voilant ainsi les bons résultats des modèles restants. Le stacking essaye de corriger cette lacune en utilisant un algorithme ou méta-modèle qui pourrait estimer les performances des modèles de base et ainsi combiner leurs résultats intelligemment.

Pour son propre entraînement, le méta-modèle utilise les résultats (classes) générés par les modèles de base. Malheureusement, la simple combinaison des classes prédites par les modèles de base ne permet pas un bon apprentissage du méta-modèle. Cela conduit à la domination du modèle de base le plus performant et le plus surentraîné et de ce fait ne garantit pas une bonne performance sur de nouvelles données. Une première solution est de diviser les données initiales en deux. La première partie des

données est utilisée pour l'apprentissage des modèles de base et la deuxième partie est utilisée pour l'apprentissage du méta-modèle. Comme la deuxième partie des données n'a pas été utilisée pour l'apprentissage des modèles de base, leurs performances vont diminuer lors de leurs utilisations pour l'apprentissage du méta-modèle évitant ainsi la domination d'un seul modèle. Une fois l'apprentissage du méta-modèle terminé, les modèles de base peuvent être ré-entraînés sur la totalité des données augmentant ainsi leurs performances. L'inconvénient d'une telle approche est que le méta-modèle n'est pas entraîné sur la totalité des données.

Une deuxième solution est d'utiliser la même technique décrite plus haut pour le bagging, utilisée aussi dans la validation croisée des modèles. C'est à dire que le méta-modèle utilisera la technique du bagging, cela lui permettra de s'entraîner sur toutes les données de départ. L'utilisation du bagging augmente considérablement le temps d'apprentissage total, parce que les modèles de base doivent s'entraîner et être testés sur chaque échantillon généré par le bagging.

Pour la génération du méta-modèle, il est recommandé d'utiliser un algorithme simple car la classification se fait au niveau des modèles de base, le méta-modèle ne fait que combiner les résultats.

L'extension du stacking à la prédiction des valeurs numériques est possible et triviale, il suffit de se limiter pour la génération des modèles de base à des algorithmes qui supportent les valeurs numériques et remplacer les attributs représentant les classes dans les modèles de base et le méta-modèle par des attributs de type numérique.

11. CONCLUSION

Dans ce chapitre, nous nous sommes intéressés au processus d'extraction de la connaissance que nous avons pris soin de distinguer du forage de données qui n'est qu'une étape parmi d'autres de tout le processus. Si les quatre phases du KD sont connues et admises par la communauté scientifique, ses étapes comme projet global mis en œuvre dans une entreprise sont peu étudiées et pas encore standardisées. Des efforts dans ce sens sont à faire dans l'avenir. Nous avons relevé certains aspects qui sont très importants pour le KD. Le problème de l'éthique, le traitement des valeurs erronées, la prise en charge des valeurs manquantes, la sélection des attributs, la discrétisation des valeurs, la réduction des données par échantillonnage, l'estimation des erreurs et l'évaluation des résultats sont des aspects qui nous ont interpellés tout au long de notre projet et qui nous a semblé utile d'en rappeler la théorie et les principes. Nous avons décrit, sans trop de détails, les principales techniques de datamining. Les variantes et les différences d'appellations de toutes les méthodes existantes rendent leur recensement des plus ardus. Nous avons aussi discuté des combinaisons de modèles lesquels apportent une plus grande pertinence des résultats moyennant des ressources matérielles et temporelles conséquentes. Pour finir, rappelons que le datamining fait appel à plusieurs domaines très proche de l'intelligence artificielle, nous pouvons citer entre autres, l'analyse de données, les bases de données, l'apprentissage, les statistiques, les systèmes à base de règles, les réseaux de neurones, la visualisation,...Cette pluridisciplinarité et la

diversité des domaines auxquels il touche, expliquent l'actuel engouement de la recherche pour le forage de données.

Chapitre III

ENTREPÔT DE DONNÉES

1. INTRODUCTION

Les systèmes informatiques traditionnels ont pour but de traiter instantanément les données qu'ils reçoivent, procédant automatiquement à une mise à jour des fichiers du système de gestion de base de données. Le traitement de l'information est considéré comme une transaction que doit réaliser le système informatique, d'où l'utilisation des termes *«transactionnel»* et *OLTP* pour *On Line Transaction Processing* ou traitement transactionnel en ligne.

On oppose souvent le traitement transactionnel en ligne au traitement analytique en ligne «OLAP», qui, lui est plus dédié à l'aide à la décision et regroupe des techniques informatiques d'analyse multidimensionnelle appliquées aux grandes bases de données et mis à la disposition des décideurs de l'entreprise pour des prises de décisions éclairées. L'entreposage de données ou le datawarehousing est à la base de l'OLAP. Ce dernier n'est, donc, pas une extension ou une amélioration de l'OLTP, mais se sont bien deux modes de travail interactifs complètement différents. Le *tableau 04* compare les deux modes.

Caractéristiques	OLTP	OLAP
Intérêts	Déclin de la recherche au début des années 90	Début de la recherche dans les années 80
Opérations	Mise à jour ponctuelle (lignes/Transaction)	Analyse et navigation dans toute la base de données
Quantité d'information échangée	Peu	Importante (version historique)
Ancienneté des données	Récente	Historique
Type d'accès	Lecture / Écriture	Lecture
Orientation	Ligne	Multidimensionnelle
Taille	Jusqu'à quelques Giga octets	De quelques Giga à ~1 Téra octets

Tableau 04 : Comparaison entre les deux modes OLAP et OLTP

2. ÉVOLUTION DES TECHNIQUES D'ENTREPOSAGE

2.1. Infocentre

Les infocentres sont une réponse, née au début des années 80, à la grande différence entre les deux systèmes OLAP et OLTP, différence dans la structure et les modèles. C'est une technologie consistant à récupérer des parties de base de données sur une seule machine pour les traitements décisionnels. C'est le premier genre de base de données de type entrepôt de données. Le principe des infocentres est d'intégrer des vues de multiples sources de données et de les exploiter pour le décisionnel. Il peut être définit comme suit :

Structure informatique dans laquelle sont collectées puis centralisées les données pertinentes à la prise de décision et est doté d'interfaces conviviales et simples pour permettre à des non spécialistes d'extraire les renseignements qu'ils désirent.

L'information dans un infocentre est volatile et sert à une prise de décision ponctuelle, ce qui le différencie de l'entrepôt de données que nous définirons plus loin.

2.2. Le Middleware

La dispersion de l'information est le principal handicap auquel se heurte une entreprise pour une bonne prise de décision. Très souvent, les applications sont développées par domaines d'activité tels que finances, commercial, stock,... Pour améliorer ses performances et augmenter sa productivité, une entreprise doit mettre à la disposition de ses employés et surtout de ses décideurs des informations dédiées à un intérêt donné. Celui-ci peut porter sur un produit ou tout autre sujet portant un intérêt d'analyse pour l'entreprise (Client, Consommateur, Vendeur, Produit, Activité,...).

Le middleware est un ensemble d'outils logiciels installés sur un seul poste client à partir duquel on peut accéder à toutes les données de l'entreprise par la technique du **Data Pull**. La solution middleware n'est pas efficace et peu envisageable sur les gros systèmes pour les raisons qui suivent :

- *Indisponibilité des middlewares sur le marché*
- *Taille et puissance du poste client nécessaires pour un tel ensemble d'outils*
- *Temps d'accès aux données distribuées*
- *Pénalisation des tâches de production*

2.3. L'entrepôt de données

Le but est de regrouper par sujet les informations disséminées au sein d'une même base de données par la technique du **Data Push** pour, d'une part, diminuer le temps d'accès aux données et d'autre part ne pas pénaliser les applications de production. La nouvelle base de données, ainsi constituée, facilite la sauvegarde de l'historique des données ce qui permet au client de faire des analyses multiples du présent et surtout du passé. Les outils de l'OLAP et ceux récents du datamining sont utilisés pour, à un premier niveau, analyser les données et prendre, à la lumières de ces analyses, les bonnes décisions et, à un deuxième niveau, modéliser les données afin d'en extraire des connaissances. On peut résumer les avantages de l'approche datawarehouse par rapport au middleware dans les points suivants :

- *Amélioration des performances du système*
- *Non perturbation des appels OLTP*
- *Diffusion des données à des moments choisis*

Un entrepôt de données est un ensemble de matériels et de logiciels informatiques réparties sur trois niveaux. Son architecture fonctionnelle est schématisée par la *figure 09*.

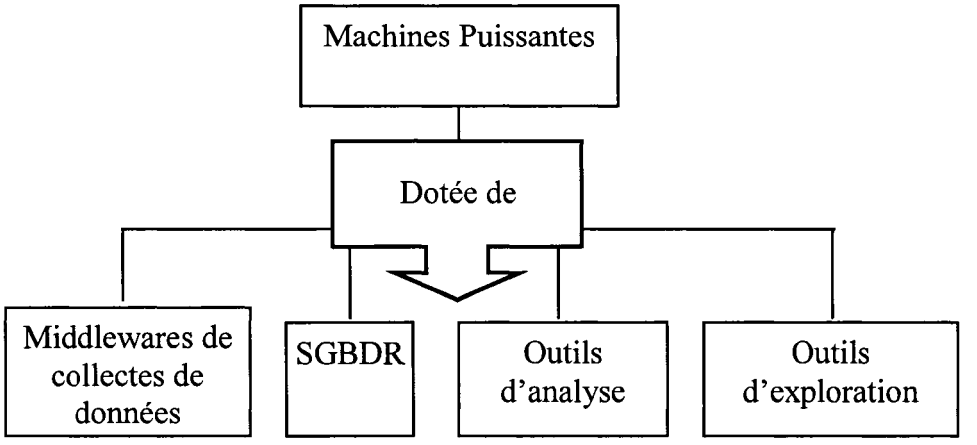

Figure 09 : Représentation schématique des composants matériels et logiciels d'un entrepôt de données

Un entrepôt de données est un ensemble de données historisées constitué par extraction à partir de bases ou de fichiers, organisé par sujets d'intérêts, consolidé dans une base de données unique, géré dans un environnement de stockage particulier, et aidant à la prise de décision dans une entreprise.

Extraction des données :

L'extraction des données à partir d'une base de données opérationnelle est réalisée par un composant logiciel appelé moniteur, selon deux approches différentes :

Approche Push : Le moniteur détecte automatiquement les mises à jours sur les bases de production et les envoie vers le datawarehouse. L'approche est utilisée si les nouvelles données sont jugées importantes pour les modèles à construire et sur la prise de décision en découlant. Son impact sur les systèmes de production doit être bien étudiée pour éviter d'éventuelles perturbations. L'approche est à considérer si le chargement des données est accompagné de calculs lourds d'agrégats, de données dérivées et de résumés de données.

Approche Pull : Le moniteur est activé périodiquement pour prélever les mises à jours des bases de production. L'approche est la plus utilisée parce que plus pratique. Elle permet d'éviter la perturbation des systèmes de production en choisissant les périodes et les durées de chargement les plus appropriées. La différence de la mise à jour entre les données de l'entrepôt de données et des bases de production n'est généralement pas pénalisante pour la prise de décision.

Fusions des données :

On distingue deux types de fusions, le chargement initial de l'entrepôt de données et la mise à jour périodique en fonction des modifications des bases de production. La fusion consiste en l'intégration des données en provenance des différentes bases de données de production et leur stockage dans le datawarehouse.

Exploitation des données :

L'exploitation se fait à deux niveaux. Le premier est l'analyse de données telles que l'analyse en tendances par des courbes et l'aide à la décision et le deuxième est l'exploration des données pour la découverte de connaissances.

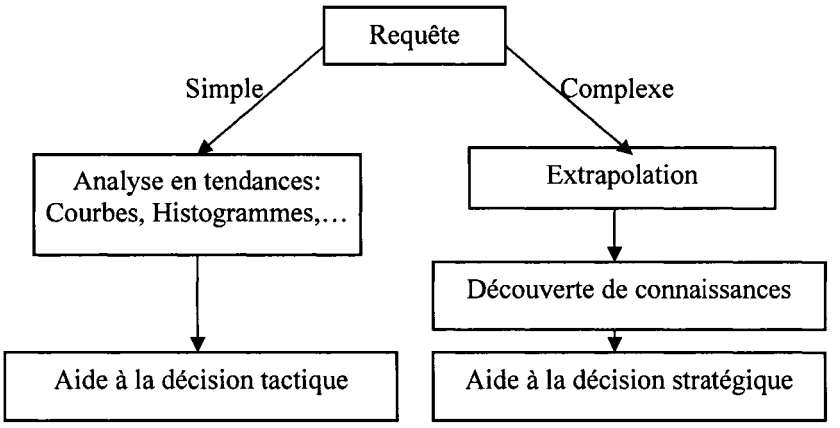

Figure 10 : Exploitation des données par des requêtes dans un Entrepôt de Données

2.3.1. Kit d'alimentation de l'entrepôt de données

Comme le montre la *figure 11* le kit d'alimentation du datawarehouse est une famille de composants

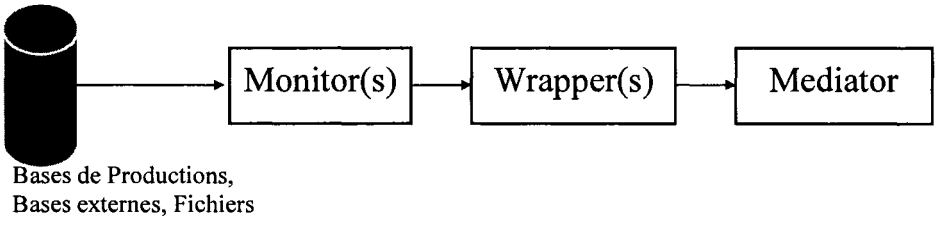

Figure 11 : Le kit d'alimentation de l'entrepôt de données

logiciels distribués entre les bases de production et le datawarehouse. Leur rôle est d'assurer la cohérence entre les informations du datawarehouse et celles des bases de production. Pour cela le middelware de collecte de données va détecter les mises à jour effectuées dans les bases de production.

Cette tâche est assignée au moniteur. Ensuite, un autre composant appelé adaptateur s'occupe de préparer l'intégration des données dans le datawarehouse en les convertissant aux formats adéquats. Avant toute sauvegarde, les données doivent être transformées, réorganisées et filtrées surtout si elles proviennent de sources multiples. Le dernier composant, appelé médiateur, s'occupe de la fusion de sources de données.

Le moniteur (Monitor):

Intégré à chaque source, il détecte les mises à jour locales, et repère les données à envoyer.

1. Il est alerté par le SGBD de la base de production si celle-ci dispose d'un mécanisme déclencheur (Trigger). C'est ce qu'on appelle la technique Push
2. Il interroge périodiquement chaque base locale, ou son journal. C'est la technique Pull.

L'adaptateur de source (Source Wrapper) :

Il est capable de traduire les données (conversion dans un format commun) d'une source locale vers le modèle du datawarehouse et vice versa.

Pour cela il :

1. Décode chaque enregistrement $Delta_i$ (Isole les attributs, les transforme dans les types de données cibles, leurs change éventuellement les unités, détecte les attributs nuls,...).
2. Génère les tuples $Delta_{ij}$ résultant pour chaque table R_j.
3. Vérifie que les tuples qui satisfont les contraintes. Les rejets sont examinés manuellement.
4. Procède à d'éventuels enrichissements (date, jointure avec d'autre tables,...).
5. Procède à d'éventuelles agrégations primaires pour réduire les données transférées.

Le médiateur (Mediator) :

- Il fusionne, si nécessaire, les données de sources multiples avant de les intégrer au datawarehouse.
- Il s'appuie sur le SGBD du datawarehouse pour faire les jointures, union, sélections et agrégats.
- Il est capable de donner une vision intégrée des différentes sources de données et d'extraire par des requêtes des parties de ces vues intégrées.

Les composants décrits ci-dessus ont des fonctions qui se chevauchent les unes sur les autres. Souvent ces fonctions ne sont pas réalisées par un seul composant mais sont réparties entre les trois.

On peut toujours ramener une mise à jour à une suppression d'enregistrements éventuellement vide suivie d'une insertion d'enregistrements éventuellement vide. Si Deltai- représentent les enregistrements à supprimer et Deltai+ les enregistrements à insérer, à chaque mise à jour est associé un couple (Deltai-, Deltai+) passé à l'adaptateur.

3. MODÉLISATION D'UN ENTREPÔT DE DONNÉES

Un modèle décisionnel peut être caractérisé par cinq axes [Fayyad96]

1. *La lisibilité :* L'utilisateur final doit être capable de comprendre le modèle.
2. *Les performances de chargement :* Le chargement de l'entrepôt de données doit être sans incidence sur le système transactionnel. La période et la fréquence de chargements doivent être choisies avec minutie et en concertation avec les administrateurs du système transactionnel.
3. *Les performances d'exécution des requêtes :* Les tailles phénoménales que peuvent rapidement prendre les entrepôts de données, exigent souvent de leurs administrateurs de calculer des agrégats et des données dérivées les plus susceptibles d'être utilisés. Comme il est difficile de prévoir toutes les requêtes émises par les utilisateurs, il devient difficile d'appliquer les techniques d'optimisation issues du transactionnel. D'autres approches d'optimisation basées sur le chemin d'accès sont utilisées.
4. *L'administration :* L'une des tâches de l'administration est de mettre l'entrepôt de données à la disposition de l'utilisateur en lui offrant les accès souhaitables et en lui interdisant les données auxquelles il n'a pas droit. Aussi, l'administration inclut le repérage et le recensement des requêtes les plus utilisées et la maîtrise et l'industrialisation de tous les processus d'extraction.
5. *L'évolutivité :* Le développement d'un entrepôt de données est incrémental et non itératif. Chaque projet décisionnel vient s'ajouter aux projets de départ pour faire un tout.

3.1. Dénormalisation

Pour une bonne modélisation d'un entrepôt de données, la dénormalisation est nécessaire. Elle résulte principalement de la fusion de plusieurs tables et l'introduction de redondances en dupliquant certains attributs.

Il n'existe pas de technique de dénormalisation standard. L'approche est pragmatique et découle d'une analyse précise des besoins de l'utilisateur. Lors de l'analyse, il est important de déterminer les sujets (patients, visites, ...) que l'utilisateur considère comme les plus importants et sur lesquels il prévoirait porter ses études. Sujets que le concepteur de l'entrepôt de données considérera comme les éléments clés lors la construction des nouvelles tables. La détermination des redondances qu'il faut ajouter aux tables sujets est une forme de dénormalisation. L'ajout du nom du médecin traitant dans la table patient

est un exemple de redondance. Le calcul des agrégations intéressantes et fréquemment sollicitées est une technique utilisées pour l'augmentation des performances du système. Le choix des agrégats est important, car même si le système dispose d'espace disque suffisant, le calcul des agrégats peut être handicapant pour le système, surtout si les calculs se font au moment du chargement.

Le gain de performance n'est pas garanti après dénormalisation. Si le nombre de tables est petit, donc le nombre de jointures potentielles réduit, les tailles des tables augmentent considérablement rendant les jointures plus lourdes à exécuter.

3.2. Dimensions hiérarchiques

Dans un système décisionnel, une dimension peut être vue comme un système d'unités «contenants, contenus», où l'unité la plus générale est le contenant de toutes les autres unités définies sur le même axe.

> *Une dimension est un axe d'analyse regroupant des indicateurs de performance et correspondant à un sujet d'intérêt.*

La dimension temporelle est présente dans pratiquement toutes les applications décisionnelles. La géographie et la hiérarchie de produits sont des dimensions potentielles. Le chiffre d'affaire d'une entreprise peut être étudié par mois, par trimestre ou par année. Il peut, aussi, être étudié par région, par catégorie de produits ou par client. Tous ces axes d'analyse représentent des dimensions intéressantes pour l'indicateur chiffre d'affaire. Les dimensions les plus fréquemment utilisées sont :

- *La dimension pays*
- *La dimension situation géographiques*
- *La dimension structure de l'entreprise*
- *La dimension produits (qui peut être très volumineuse notamment dans le cas de la grande distribution)*
- *La dimension projets*
- *La dimension Clients*
- *La dimension Fournisseurs*
- *La dimension Temps*

Chaque unité peut être mesurée par une unité plus ou moins fine. Lors de l'exploitation, les analystes étendent ou réduisent la représentation des données suivant les axes des dimensions. Certaines dimensions nécessitent pour le besoin de leurs développements l'accès à la base de données. Ce sont généralement les dimensions non denses telles que la dimension géographie ou la dimension produits.

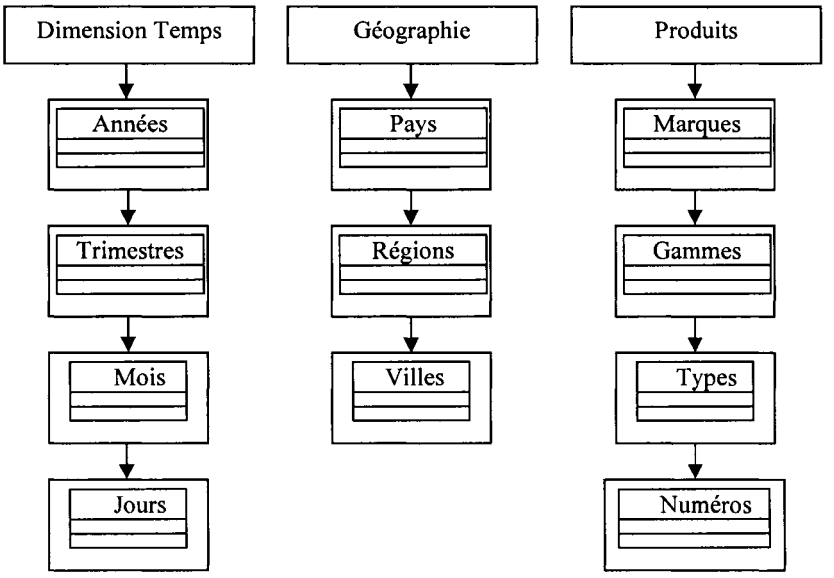

Figure 12 : Exemples de dimensions

Lorsqu'une dimension est documentée par une autre table, on peut exploiter les données de cette dernière dans le but d'affiner les analyses en agrégeant sur des attributs moins spécifiques retrouvés dans ces tables.

La dimension *N°Fournisseur* est documentée par la table *(N°F, Nom, Ville, Région)*

- Par jointure avec fournisseurs et quelques agrégations, on peut analyser les ventes selon la ville, région,... C'est ce qu'on appelle le Drilldown (Dépliage).
- Comme on peut revenir depuis les régions et villes aux fournisseurs. C'est ce qu'on appelle communément le Rollup (Pliage).

3.3. Modélisation par sujets

Les applications telles que conçues dans les bases de production ne sont pas adaptées pour être exploiter. Une modélisation par sujet s'avère nécessaire. Chaque sujet qui intéresse l'utilisateur est associé à une table gérée par le Datawarehouse. Pour ce faire il faut :

- *Isoler les données stratégiques.*
- *Déterminer les informations de détail nécessaires.*
- *Déterminer les résumés à conserver.*
- *Déterminer les méta-données décrivant les tables (nom d'attribut, unité, valeur par défaut,..., mémoriser si possible les sources de production pour chaque table, confidentialité,...)*

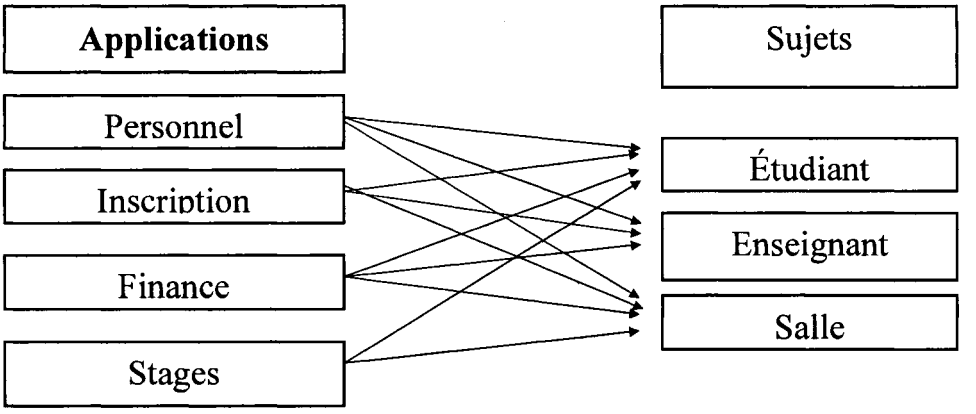

Figure 13 : Modélisation par sujets

3.4. Modélisation multidimensionnelle

La modélisation dimensionnelle est indépendante du modèle de base de données sous-jacent. Elle s'applique aux bases de données relationnelles, multidimensionnelles ou objet. Cette technique privilégie la performance et la considère comme un objectif central. La performance est déterminée par l'utilisateur. Le chiffre d'affaire et la rentabilité sont des indicateurs de performance dans le domaine commercial alors que le taux d'occupation d'un lit et la durée de séjour d'un patient sont des indicateurs de performances en milieu hospitalier. L'analyse de ces indicateurs se fait à travers les dimensions.

Modèle en étoile

Un des modèles le plus populaire est le modèle en étoile. Dans ce genre de modèles les indicateurs de base sont groupés dans une table centrale, appelée table de faits, autour de laquelle gravitent les tables de dimensions associées aux indicateurs. L'identifiant de la table de faits est une clé multiple composée de la concaténation des clés de chacune des tables de dimension. La table de fait du chiffre d'affaire a comme clé la concaténation des clés des tables Temps, Produit, Région et Client.

Le modèle en étoile part du principe que l'utilisateur est essentiellement intéressé par l'analyse des indicateurs de performance selon les diverses dimensions les caractérisant. Ceci a pour conséquence de faciliter la lisibilité du modèle par l'utilisateur et d'augmenter ces performances.

> *Lisibilité :* Les indicateurs de performances et les dimensions d'analyse sont clairement représentés par le modèle.
>
> *Performance :* La force du modèle en étoile est que les chemins d'accès à la base de données sont prévisibles. Caractéristique importante utilisée pour l'optimisation des requêtes. Comme les tables de dimensions ne sont pas très grandes, l'exécution d'une requête passe tout d'abord par des sélections sur ces tables avant l'accès aux tables de

faits. Ce qui réduit considérablement les enregistrements parcourus. Une bonne indexation des tables de faits est toutefois nécessaire pour que les performances du système ne dépendent que du volume de données attendues en sortie.

Modèle en flocon (Snowflake)

Le modèle en flocon est une extension du modèle en étoile où les dimensions (branches) sont décomposées en hiérarchies. La décomposition est en fait une normalisation des tables de dimensions. La table de dimension Temps peut être, par exemple, éclatée en sous-tables *Année*, *Mois*,... ou en *sessions, Automne, Hiver, Été.*

Le modèle en flocon réduit les tailles des tables de dimensions en les normalisant et aussi formalise la notion d'hiérarchie, ce qui peut faciliter les analyses. Mais, par contre, il peut rendre complexe la gestion de la base de données.

3.5. Faiblesse de la modélisation multidimensionnelle

L'extensibilité

Une des faiblesses du modèle multidimensionnel est son extensibilité difficile. Comme les dimensions choisies sont faites sur la projection de la future utilisation du système, toute nouvelle orientation dans les objectifs peut engendrer l'ajout de nouvelles dimensions. Cet ajout supposerait le retour au modèle transactionnel de départ, l'ajout de la table représentant la dimension souhaitée et finalement, la mise à jour de la table des faits en incluant les clés de la nouvelle dimension. Cette opération, apparemment toute simple, peut augmenter de façon significative la taille de la table de faits; taille qui peut être multipliée par un facteur de 100 ou plus.

Orientation objectif

Le fait que le modèle soit construit sur la base d'indicateurs de performance et d'axes d'analyse, le rend étroitement dépendant de ceux-ci. La résolution d'un autre problème dont les indicateurs de performance et les dimensions sont différents de l'ancien modèle est difficilement faisable. L'évolutivité du modèle s'en trouve, ainsi, compromise.

3.6. Le dataMart

Le datamart ou mini-entrepôt est, comme son nom l'indique, un entrepôt de données de taille réduite. Il correspond à une base de données spécialisée se rapportant à un secteur d'activité particulier de l'entreprise ou à un métier qui y est exercé (commercial, marketing, comptabilité, etc.).

Vu sa relation directe avec l'analyse, nombre de spécialistes pensent que le modèle multidimensionnel est plus adapté au datamart qu'à l'entrepôt de données. Ce dernier est vu plus comme un entrepôt dans lequel toutes les données de l'entreprise sont historisées, sans être structurées dans un but précis d'analyse. Alors que le datamart est implanté au niveau des services ou départements et est structuré pour répondre aux besoins d'analyses de ces derniers. Cette approche pose un problème de double

stockage de l'information dans le datamart et l'entrepôt de données. Des solutions devront être proposées. Néanmoins, si l'idée est généralisée à toute l'entreprise, les données présentes dans les datamarts peuvent être supprimées de l'entrepôt, ce dernier ne servant plus à sauvegarder toutes les données mais à jouer le rôle de passerelle de communication entre les différents datamarts et le système transactionnel.

4. CUBE MULTIDIMENSIONNEL

Il ne suffit pas de disposer des données regroupées par sujet dans un datawarehouse, mais il faut aussi pouvoir exploiter ces données. L'une des approches adoptées est la modélisation multidimensionnelle des données grâce à un cube de données. C'est une nouvelle méthode pour faciliter l'analyse de données appliquée au datawarehouse.

Définition :

Un cube de données est une représentation de N attributs extraits d'une table sous forme d'un cube. N-k attibuts composant les dimensions le long desquelles les groupements sont possible, les k autres étant des mesures résultats de fonctions d'agrégations (généralement k = 1).

La *figure 14* illustre la modélisation multidimensionnelle. Elle représente un cube de données sur l'attribut quantité selon les axes *N°Pro, N°Fou* et *Date* à partir de la table *Ventes (N°V, N°P, N°F, Date, Quantité, Prix)*. Par projection, on obtient la table: *VentesCube (N°P, N°F, Date, Quantité)*. Le cube n'est que la représentation de la vue *VentesCube*.

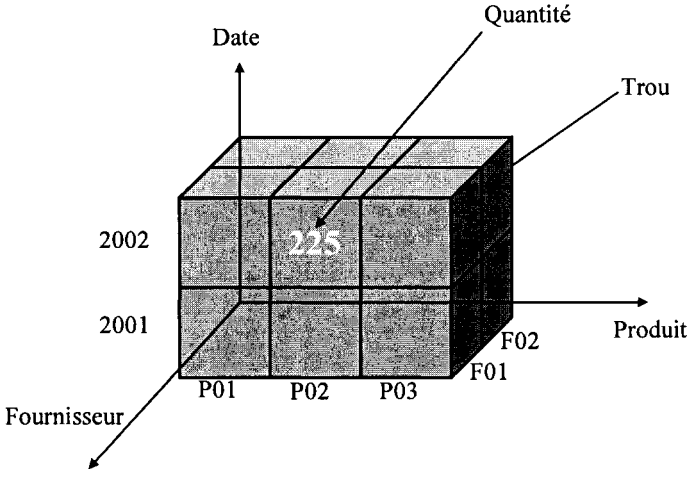

Figure 14 : Exemple de cube de données

Différents choix d'unités sont possibles sur chacun des axes. La valeur de la variable analysée appelée indicateur, figure dans chaque cube unitaire si elle est définie. Dans notre illustration de la *figure 14* à la date 2002, pour le produit P02 et le fournisseur F01, la quantité est 225.

Comme la plupart des valeurs ne sont pas définies le cube est plein de trous correspondant à des valeurs non existantes. En fait le cube n'est que du vide parsemé de quelques valeurs disséminées.

Vue d'un cube :

La vue d'un cube est définie à partir d'un cube de données par agrégation des quantités selon un sous ensemble d'attributs.

Les vues sont obtenues par sélection selon une dimension et des projections avec agrégation des mesures par groupements selon les dimensions éliminées. L'aspect visuel des résultats des coupes ainsi faites facilite grandement les analyses.

Pour un cube de dimension k, il existe 2^k vues obtenues par agrégation. Les vues sont organisées en un treillis. La *figure 15* représente le treillis issu de l'exemple vu ci-dessus. La fonction d'agrégation *SUM* et éventuellement *COUNT* sont utilisées pour rendre les vues auto-maintenables.

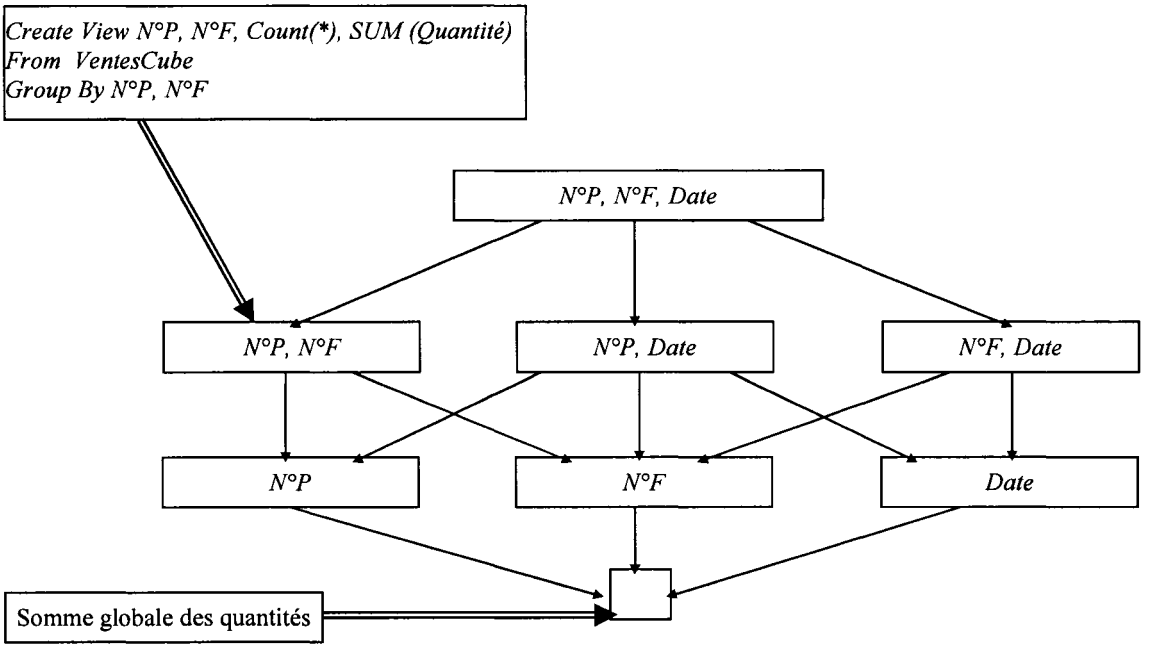

Figure 15 : Treillis de vues d'un cube de données

4.1. Calcul du cube

Pour le calcul du cube un nouvel opérateur a été défini par gray [Gray95] en 1995. C'est une généralisation à N dimensions de l'opérateur group by, il a la forme suivante :

Select $B_1, B_2, \ldots B_k$, *AGG(A)* Où B_i attributs de *R* constituants les dimensions du cube
FROM R *A* attribut agrégé
CUBE BY $B_1, B_2, \ldots B_k$

Toutes les applications d'analyse de données multidimensionnelles se basent sur les agrégations sur plusieurs dimensions. Le calcul des agrégats peut provoquer des étranglements pour ces applications. L'opérateur cube a besoin du calcul des group bys sur toutes les combinaisons possibles de la liste d'attributs, donc pour autant d'agrégations. Le problème est résolu en classant les opérateurs group bys dans une hiérarchie. Ces algorithmes ne sont que l'adaptation des méthodes de groupement basées tri ou basées hashing.

La vitesse d'exécution est la plus grande contrainte des applications OLAP et ceci dans le souci de rendre les analyses interactives (réponse en quelques secondes). Le calcul préalable des agrégats à différents niveaux de détails et différentes combinaisons d'attributs s'avère nécessaire. La vitesse et le coût sont aussi critiques pour ces calculs au préalable dans la mesure ou ils pèseront lourds sur les fréquences de mise à jour des agrégats.

Supposons que dans une application de la grande distribution on ait une table des transactions qui contient les attributs *Produit (P), Date (D), Client (C) et Ventes (V)*. Toute la collection des requêtes d'agrégats peut s'exprimer de la sorte :

Select P, D, C, Sum (V)
FROM Transactions
CUBE BY P, D, C

Cet exemple possède $2^3 = 8$ group bys *PDC, PD, PC, DC, P, D, C* et *all* (tout ou group by vide). Quelles solutions apportées à la complexité du calcul du cube? C'est la question à laquelle répond la littérature [Agarwal96] par la proposition de deux types de solutions :

Première solution (Algorithme Naturel)

Mémoriser le cube et ses vues comme un tableau à k dimensions. Chaque groupement GROUP BY est calculé de façon indépendante. Ce qui veut dire réécrire la requête à l'aide de 8 requêtes Group By et faire le calcul pour chacune d'elle. L'inconvénient de l'algorithme naturel est qu'on a 2^k groupements possibles. Le temps de calcul est très élevé quand k est grand.

Deuxième solution (Algorithme basé sur les parents directs)

Éviter les calculs répétés en exploitant les treillis. Il faut maximiser la réutilisation des résultats des cubes précédents pour calculer des cubes plus fins. La fonction d'agrégation du cube change aussi la nature du

problème. Les fonctions distributives comme *somme, compte, maximum, minimum* sont calculables par agrégation de résultats sur sous-ensembles. Les fonctions algébriques telles que la moyenne peuvent être ramenées au cas distributif en maintenant un compteur. Les seules fonctions à poser problème sont les fonctions holistiques comme la médiane. Dans ce cas, on n'a pas le choix que de calculer les 2^k groupements indépendamment.

4.1.1. L'algorithme PipeSort (Méthode basée sur le tri)

Il est basé sur le partage du tri lors du calcul des Group Bys. Par exemple, si les données sont triées selon les attributs A, B, C et D dans cet ordre, alors on peut calculer les Group Bys *ABCD, ABC, AB* et *A* sans tri additionnel. Cette approche ne prend pas en compte la taille des données en mémoire. Si *BDA*, duquel on peut aussi calculer *AB*, est moins volumineux que *ABC*, n'est-il pas plus judicieux de calculer *AB* de *BDA*? Il faut définir une planification globale pour déterminer comment les Group Bys sont calculés, de quel parent et dans quel ordre.

L'algorithme PipeSort combine les deux optimisations, distribution des tris et le plus petit parent pour avoir un coût minimal. L'optimisation des accès disque est assurée par l'exécution de plusieurs Group Bys dans un modèle en Pipeline. Par exemple, pour calculer les Group By *ABCD, ABC, AB* et *A*, au lieu de calculer chaque Group By séparément on pourrait les calculer selon un modèle pipeline. Chaque fois qu'un tuple *ABCD* est calculé il est propagé vers le haut pour calculer *ABC* dont le résultat est propagé pour calculer *AB* et ainsi de suite.

- Chaque pipeline est un ensemble de Group By calculé par un seul passage sur les données d'entrées.
- Un seul tuple pour chaque Group By est sauvegardé en mémoire.

4.1.2. L'algorithme PipeHash (Méthode basée sur le Hachage)

Le but principale du hashage est la gestion de l'allocation mémoire de plusieurs tables pour optimiser l'utilisation de la mémoire vive et les accès disque. Par exemple, si les deux tables *AB* et *AC* peuvent être chargées en mémoire, alors elle peuvent être calculées en un seul passage sur *ABC* et quand *AB* est calculé, on peut calculer *A* et *B* tant que *AB* est encore en mémoire.

Pour k = N à 0 faire

Pour chaque Group By g de k+1 Attributs faire

Calculer en un seul passage de g tous les Group Bys de k Attributs pour lesquels g est le plus petit parent ;

Sauvegarder g sur le disque et détruire la hash table de g;

Fin Pour

Fin Pour

Si les données sont trop grandes pour être chargées en mémoire, on les partitionne sur un ou plusieurs attributs *A* et puis tous les Group By qui contiennent *A* peuvent être calculés par des groupements

indépendants de chaque partition. Les coûts de partitionnement sont distribués sur tous les Group Bys contenant les attributs partitionnés.

4.2. Les opérateurs

Drilldown (Dépliage) :

Extension d'une dimension du cube en la remplaçant par une dimension à grains plus fins, soit en allant du global vers le détails.

Dans notre exemple précédent du cube *(Fournisseur, Année, Pays)*, on peut appliquer le Drilldown sur les axes suivant :

- Axe du temps : *Drilldown (Année Mois), Drilldown (Année Mois Jour)*
- Axe géographique : *Drilldown (Pays Région), Drilldown (Pays Région Ville)*
- Axe fournisseurs : *Drilldown (Fournisseurs Produits)*

Rollup (Pliage):

Réduction d'une dimension du cube en la remplaçant par une dimension à grain plus large, soit en allant du détail vers le global.

Dans notre exemple précédent du cube *(Produits, Date, Ville)*, on peut appliquer le Drilldown sur les axes suivant :

- Axe du temps : *Rollup (Année Mois), Rollup (Année), Rollup ()*
- Axe géographique : *Rollup (Pays Région), Rollup (Pays), Rollup ()*
- Axe produits : *Rollup (Fournisseurs)*

Le Drilldown, Rollup ne sont possibles entre deux attributs A et B que s'il y a une dépendance fonctionnelle entre ces deux derniers.

Slice (Coupe):

Sélection de tranches de cube par des prédicats selon une dimension. Il s'agit de sélectionner des portions du cube en filtrant une dimension selon une valeur ou une plage de valeurs. La dimension filtrées est alors translatée et éventuellement tassée si des trous apparaissent, en cas de filtrage sur plusieurs valeurs non successives par exemple.

Dans l'exemple précédent, on peut appliquer le Slice sur le cube *(Produits, Date, Ville)* selon les axes suivant :

- Axe du temps : *Slice (20/09/2001), Slice (>1998 and <1999)*
- Axe Produit : *Slice ('Ordinateurs')*
- Axe Géographique : *Slice ('Trois Rivières'), Slice ('Trois Rivières' or 'Montréal')*

La combinaison des opérateurs Drilldown, Rollup et Slice forme l'algèbre des cubes de données. Il n'y a aucune restriction autre que la nature des données sur les combinaisons possibles. Par exemple, avec le cube (N°P, Date, Région) on peut faire le calcul suivant :

Slice ('Trois Rivières')
[Drilldown (Ville)
[Rollup (Année,Mois)
[Slice (>1998 and < 1999) [Cube]]]]

Sur le même cube on veut étudier sur *l'année 98*, par période d'un mois, dans la ville de Trois-Rivières, les quantités de produits vendues.

Les outils commerciaux définissent d'autres opérateurs de modélisation du cube, opérateurs pas ou peu étudiés par les scientifiques. Je peux citer :

Drill through :

Le Drill through consiste à visualiser une même information sous l'angle de plusieurs dimensions. On pourrait, par exemple, souhaiter à partir de la visualisation du chiffre d'affaires d'un point de vente par produits et pour un mois déterminé, obtenir la visualisation de la même information mais pour un autre point de vente.

4.3. Mise à jour du cube

Dans le cas des vues auto-maintenables, il est très aisé de faire la maintenance des vues de façon efficace. L'idée des algorithmes est d'apporter les mises à jour par paquets à chaque transaction ou de façon périodique (fin de journée, fin de semaine, décision interne,...) [Mumick97]. Des compteurs sont utilisés pour rendre les vues auto-maintenables. La mise à jour se fait en deux étapes :

1. Calcul d'une table résumé des modifications à apporter, qu'on peut traduire par un RésuméDelta. Ce résumé est calculé à partir des tuples à insérer et à supprimer dans la table de base. Il a le même schéma que la vue, où la clé correspond aux dimensions du cube et comporte un champ compteur dont la valeur est incrémentée de un pour chaque tuple inséré dans la table de base et décrémentée de un pour chaque tuple supprimé de la table. Le compteur représente le delta à apporter au compteur de la vue. Les agrégats sont calculés pour chaque modification et les résultats sont mémorisés dans la table.
2. Mise à jour concrète de la vue à partir de la table RésuméDelta. Cela est fait par le balayage du résumé tuple à tuple, puis l'accès à la vue par l'intermédiaire d'une clé. Une fois positionné, la mise à jour est effectuée selon la valeur du RésuméDelta et le compteur indiquant le nombre de

tuples source est mis à jour à l'aide du compteur du résumé. Si la nouvelle valeur du compteur est zéro, le tuple est supprimé. Puis les agrégats sont mis à jour.

Exemple:

Reprenons l'exemple précédent [Gardariniі99] qui illustre la mise à jour concrète de la vue *VentesCube (N°P, N°F, Date, Quantité)* à partir des modifications apportées à la table *Ventes (N°V, N°P, N°F, Date, Quantité, Prix)*. Les mises à jour sont de trois types, une modification (traduite par une insertion et une suppression) qui concerne le produit N°10, une suppression portée sur le produit N°12 et une insertion portée sur le produit N°11.

	N° Vente	N° Produit	N° Fournisseur	Date	Quantité	Prix
DELTA+	1	10	100	06/2001	18	98
	2	11	200	01/2002	15	227

	N° Vente	N° Produit	N° Fournisseur	Date	Quantité	Prix
DELTA-	1	10	100	06/2001	7	98
	3	12	200	05/2002	25	111

	N° Produit	N° Fournisseur	Date	Compteur	Quantité
Résumé Delta	10	100	06/2001	0	11
	11	200	01/2002	1	15
	12	200	05/2002	-1	-25

+

	N° Produit	N° Fournisseur	Date	Compteur	Quantité
VenteCube	10	100	06/2001	2	12
	11	200	01/2002	1	35
	12	200	05/2002	1	25

=

	N° Produit	N° Fournisseur	Date	Compteur	Quantité
VenteCube	10	100	06/2001	2	21
	11	200	01/2002	2	50

5. NUCLEUS SERVER[7]

Inventée par Dr. Ted Glaser, ancien d'IBM, d'MIT et de la NSA, nucleus technologie profite de sa façon de stocker les données pour améliorer ses performances. L'architecture de stockage des données est une nouvelle façon de faire diamétralement opposée à tout ce qui se fait dans les autres SGBDR. Raison qui fait que cette approche ne pourra jamais être intégrée par les autres constructeurs dans leurs SGBDR, sauf s'ils décident d'opérer des changements radicaux dans leurs approches de stockage de données.

5.1. Stockage de données

Dans une table relationnelle ordinaire, comme la table des patients ci-dessous, les lignes ou enregistrements représentent des informations de divers types concernant chaque patient alors que les colonnes représentent un type particulier de données sur tous les patients. Le stockage d'une telle table se fait comme une suite d'enregistrements considérés comme identiques. Le principe est simple et facilement compréhensible. Malheureusement, les Entrées/Sorties, l'indexation quasi obligatoire pour les grands volumes de données et la maintenance qui en découle sont de véritables handicaps à surmonter. Le besoin d'un administrateur de base de données pour gérer et dénormaliser les bases de données afin de maintenir une bonne performance du système ou l'améliorer après des efforts coûteux est la preuve de la relative efficacité du modèle relationnelle.

Patient	Visite	Sexe	Occupation
12356	FL20154	M	Fermier
36984	FL25847	M	Peintre
25874	FL96387	F	Chimiste
56211	FL21458	M	Fermier
58285	FL15849	F	Plombier

Tableau 05: Table des patients

Nucleus, grâce à sa technique unique de stockage de données, évite tous ces problèmes sans pour autant diminuer les performances. Tout au contraire, ces dernières sont augmentées à un moindre effort. L'architecture de stockage des données se base sur deux principes:

- *Le stockage des tables de données se fait par colonne.*
- *Les valeurs des données sont séparées de leurs utilisations dans les tables.*

En utilisant deux techniques :

- *Des vecteurs binaires pour séparer les valeurs de leurs utilisations dans les tables.*
- *Les vecteurs binaires sont stockés et utilisés dans un format compressé.*

[7] La majorité de l'information et des exemples sont tirés d'un document de travail de Sand Technology intitulé Collaborators and Partners

Les colonnes

La séparation des colonnes d'une même table apporte un réel avantage puisqu'il ne sera plus nécessaire de charger en mémoire tout un enregistrement pour lire la valeur d'un champ spécifique. Si dans notre exemple, on ne s'intéressait qu'aux occupations des patients, seule la colonne concernée est chargée en mémoire.

Patient	Visite	Sexe	Occupation
12356	FL20154	M	Fermier
36984	FL25847	M	Peintre
25874	FL96387	F	Chimiste
56211	FL21458	M	Fermier
58285	FL15849	F	Plombier

Tableau 06: Séparation des colonnes dans la table des patients

Séparation des valeurs de leur utilisation

Pour séparer les valeurs de leur utilisation, Nucleus se sert, en plus du vecteur binaire, des identificateurs d'uplets (tuple identifier TID) et des identificateurs de valeurs (VID). Cela a l'avantage de remplacer les comparaisons des valeurs littérales par des opérations binaires très rapides à effectuer. La tokenisation se fait en trois étapes:

1. Ajouter à chaque table un identificateur de tuple (TID). Un TID est tout simplement le numéro de ligne qui permettra une identification unique de chaque tuple.

TID	Patient	Visite	Sexe	Occupation
1	12356	FL20154	M	Fermier
2	36984	FL25847	M	Peintre
3	25874	FL96387	F	Chimiste
4	56211	FL21458	M	Fermier
5	58285	FL15849	F	Plombier

Tableau 07: Ajout de l'identificateur de tuple dans la table des patients

2. Pour chaque colonne, assigner à chaque valeur distincte un entier unique comme identificateur (VID). Les colonnes de type entier peuvent utiliser leurs valeurs propres comme identificateurs. Cela procure à chaque colonne un ensemble de valeurs qui est en fait un sous ensemble du domaine des valeurs de la colonne.

Occupation	VID
Fermier	1
Peintre	2
Chimiste	3
Plombier	4

Tableau 08 : Ensemble des valeurs de l'attribut Occupation

3. Créer des paires composées de TID et de VID associés à chaque table, pour faire correspondre à chaque ligne de la table identifiée par son TID ses valeurs associées aux attributs identifiées par leurs VID. Le *tableau 09* illustre la tokenisation de la colonne *Occupation* de la table *Patient*.

TID	VID
1	1
2	2
3	3
4	1
5	4

Tableau 09 : Exemple de Tokenisation de la colonne Occupation

Représentation par la matrice binaire

Les paires de TID et VID ainsi formées sont utilisées pour construire une matrice d'incidence binaire pour chaque colonne d'une table. La colonne *Occupation* de la table *Patients* est représentée dans le *tableau 10*.

	VID			
TID	1	2	3	4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	1	0	0	0
5	0	0	0	1

Tableau 10 : Matrice binaire de la colonne Occupation

En transformant les tables en une collection de matrices binaires, Nucleus bénéficie d'une performance naturelle puisque toutes les opérations relationnelles sont réduites en des opérations binaires élémentaires sur des vecteurs binaires.

Codage des vecteurs

Si pour une colonne donnée le nombre de valeurs distinctes était d'un million, il faudrait un million de colonnes pour les représenter. Si chaque valeur était unique dans la table, chaque colonne contiendrait un bit de un et 999 999 bits de zéro, soit 128 giga octets de données. Une première remarque est que la taille de la matrice binaire peut rapidement croître en fonction du nombre de valeurs distinctes de la colonne. Pour remédier à cela, Nucleus utilise une technique de codage des vecteurs. Elle permet de compresser les vecteurs dont une valeur, soit zéro ou un, prédomine. Un vecteur d'un million de zéro peut être encodé sur 32 bits. La matrice décrite plus haut peut alors être encodée sur 01 million de vecteurs de 04 octets chacun. C'est une compression de 32 000 fois.

5.2. Traitement des requêtes

L'interrogation d'une base de données Nucleus est de 10 à 1000 fois plus rapide que dans un SGBDR traditionnel. Cette performance est le résultat du nombre limité des Entrées/Sorties que Nucleus effectue et aussi, de la façon dont est stockée l'information et traitée par les algorithmes d'interrogation de données. Pour illustrer l'exécution d'une interrogation sous Nucleus nous présentons ci-dessous un petit exemple.

Supposons qu'on veut chercher, dans la table *Patient*, tous les fermiers de sexe féminin. L'algorithme s'exécute en quatre grandes étapes.

1. Localiser le VID représentant la valeur «*fermier*» dans la colonne «*Occupation*». La recherche est rapide puisqu'elle se fait sur un ensemble ordonné de VID.
2. Localiser le VID représentant la valeur «*F*» dans la colonne «Sexe». La recherche est encore plus rapide puisqu'il n'y a que 2 valeurs possible.
3. Effectuer un ET logique sur les deux vecteurs binaires résultant des deux premières étapes. Le résultat est un vecteur binaire dans les valeurs 1 correspondent aux positions satisfaisant les deux conditions de départ.
4. Retourner toutes les rangées de la table *Patient* qui correspondent aux valeurs 1 du vecteur résultat de l'étape 3.

En comparaison avec les SGBDR traditionnels, l'algorithme décrit plus haut n'effectue aucun parcours séquentiel de la table «*Patient*» et cela sans utiliser de fichier d'index. La recherche de la valeur «*fermier*» ne se fait pas en comparant pour chaque enregistrement, le champ «*Occupation*» avec la valeur «*fermier*», comparaison entre deux valeurs alphabétiques plus longue à effectuer que la comparaison de deux VID de type entier. Les enregistrements ne sont pas chargés entièrement en mémoire, l'algorithme se contentant des colonnes de la table concernée par la requête. Cela réduit énormément le volume des Entrée/Sorties et la taille de la mémoire vive sollicitée.

Traitement des sous requêtes

Le serveur Nucleus est très performant quant au traitement des sous requêtes et des jointures, particulièrement celles qui peuvent être converties en sous requêtes dans la clause *WHERE*. Si le résultat est extrait d'une seule table, Nucleus n'effectue aucune jointure et le traitement des sous requêtes se résumera aux combinaisons de vecteurs binaires en utilisant les opérateurs logiques AND, OR et NOT. Le résultat de ces traitements est un vecteur binaire unique qui pourra être appliqué à la table interrogée. Cette qualité de Nucleus est très appréciable surtout pour les jointures externes et les sous requêtes impliquant le prédicat NOT IN. Aussi, dans les SGBD traditionnels, il est recommandé d'exécuter des jointures sur les colonnes indexées, sinon, un long pré-traitement de tri est requis, alors que dans Nucleus, n'importe quelle colonne peut être utilisée sans aucune restriction.

5.3. Le modèle de données

Nucleus utilise le modèle en étoile de représentation des données. Les tables de la base de données relationnelle sont dénormalisées et transformées en des tables de fait et des tables de dimensions. La table de fait représentant le centre du modèle et les tables de dimensions gravitent autour. Cela réduit considérablement le nombre de tables et par conséquent de nombre de jointures. Dans les SGBD traditionnels la réduction du nombre de jointures n'augmente pas forcément les performances du système. Les tailles énormes des tables utilisées et l'augmentation du nombre des Entrées/Sorties pour le traitement des jointures en est la cause. Pour remédier à ces problèmes, certains SGBD utilisent des techniques d'optimisation spécialisées, comme les tables d'index de pré-jointure. Avec le serveur Nucleus, des performances appréciables sont atteintes sans optimiseurs spécialisés ni tables d'index additionnelles, l'indexage intrinsèque des tables constituant un puissant outil de jointure.

La réduction du nombre de tables introduit explicitement des redondances de données dans une même table. Plus les valeurs de données se répètent plus le facteur de compression des vecteurs binaires est grand. Cette caractéristique conduit à une meilleure prise en charge des jointures de grandes tables.

L'augmentation de la taille globale des données est un autre inconvénient introduit par le modèle en étoile. En effet, l'ajout de dimensions implique l'ajout de clé pour chaque dimension dans la table des faits. Des index additionnels sont ajoutés pour chaque clé. Les agrégats et les données dérivées pré-calculés, compliquent les choses en ajoutant des données supplémentaires. Comme expliqué précédemment, Nucleus n'a d'une part pas besoin d'index et d'autre part, la grande redondance des données dans les tables de faits n'impliquerait qu'une légère augmentation de la taille des vecteurs binaires représentant ces données. Le modèle en étoile est très bien supporté par la technologie Nucleus.

5.4. Performances

Mémoire vive

Pour accroître la vitesse d'exécution d'une application, il est recommandé d'éviter les Entrée/Sortie et de garder toutes les données nécessaires en mémoire vive aussi longtemps que l'application en a besoin. C'est ce que tentent de faire les systèmes dits *"in-memory"*. Plus le volume des données est grand plus le système a besoin de mémoire pour stocker toute l'information et ainsi éviter une diminution brutale de ses performances. Nucleus a une architecture de traitement qui optimise l'utilisation du principe de la mémoire virtuelle. Il profite de l'organisation interne de ses données qui est similaire à celle de l'adressage d'espace virtuel. Au lieu d'avoir toute l'information en mémoire, nucleus procède par pagination, en ne chargeant que les données nécessaires. Nucleus n'a besoin, pour atteindre des performances optimales, que d'une mémoire vive équivalente à 1,5 à 3% du volume total des données entreposées dans la base de données [IBM01]. Il est alors possible de traiter les données de taille considérable d'un entrepôt de données sans disposer nécessairement d'une mémoire vive de taille conséquente. Les performances du serveur Nucleus sont le fruit de la combinaison de son architecture de stockage de données et d'un système optimisé pour la gestion de la pagination des données en mémoire.

Indexation sans index

Les vecteurs binaires sont automatiquement créés lors de l'insertion des données. Cela crée une sorte de mécanisme d'indexation intrinsèque à Nucleus. Alors que dans les SGBDR traditionnels l'indexation est une tâche distincte qui a besoin d'une gestion séparée et d'une maintenance régulière et difficile à maîtriser. Aussi, l'indexation sous Nucleus n'a pas besoin d'espace additionnel pour les fichiers d'index. Nucleus offre tous les avantages d'indexation sans générer des coûts supplémentaires d'espace de stockage et de maintenance.

Réduction de la taille de la base de données

L'architecture de stockage des données de Nucleus permet une réduction considérable de la taille des données stockées, surtout si celles-ci sont constituées de valeurs redondantes ayant une distribution favorisant leur compression. Nucleus permet une réduction du volume de données allant de 1 à 3 fois comparativement aux SGBD traditionnels. Nucleus peut aussi profiter des colonnes qui partagent le même domaine de valeurs pour leur faire partager les mêmes VID et ainsi réduire encore plus la taille de la base de données. Le non besoin d'index est un autre moyen de gain d'espace de stockage.

Vitesse d'exécution

Il est connu que les performances d'un SGBD décroissent de façon linéaire avec l'augmentation du volume des données. Pour les bases de données de quelques dizaines de giga octets, les requêtes, surtout les plus complexes d'entre elles, sont très difficiles à exécuter. L'augmentation des Entrés/Sorties est une des explications. Une des constatations les plus spectaculaires engendrée par l'architecture

particulière de Nucleus est que le temps d'exécution par enregistrement décroît en fonction des lignes de la table traitée. Cette décroissance est plus probante pour les requêtes très sélectives.

Transparence

La technologie Nucleus qui est essentiellement caractérisée par l'orientation en colonne, la tokenisation et le traitement des vecteurs binaires compressés, est totalement transparente à l'utilisateur. Toutes les opérations relationnelles sont exécutées sur des vecteurs binaires compressés. Ceux-ci n'étant jamais décodés pour la lecture des données. L'utilisateur manipule des tables relationnelles ordinaires. Nucleus s'occupe de l'interface avec l'architecture interne. Aucun travail supplémentaire n'est requis. Au contraire, l'administration d'une base de données Nucleus est beaucoup plus simple que celle d'un SGBDR traditionnel. Il n'y a ni indexation, ni groupage (clustering), ni verrouillage administratif, ni configuration de la taille des blocs, ni des partitions à gérer.

6. CONCLUSION

Le datawarehouse constitue un nouveau marché pour les constructeurs et éditeurs de logiciels. Les datawarehouses mettent à la disposition du datamining de la matière à analyser pour extraire de la connaissance. Les outils d'analyse devraient se sophistiquer. De grands projets de recherches sont menés actuellement, ils concernent les datawarehouses et les techniques de datamining.

Plusieurs défis restent à relever, tels que le développement de boîtes à outils pour construire des moniteurs et les adaptateurs pour les sources semi structurées [Widom95]. Cela peut être très utile pour enrichir les données de l'entreprise par des données externes, celles du Web notamment. L'étude de méthodologie de conception de datawarehouse n'est qu'à ses débuts. Alors que des améliorations des performances des datawarehouses par une meilleure intégration des techniques d'optimisation des requêtes est fort souhaitée. L'optimisation peut se faire sur les algorithmes des opérateurs, ainsi que sur la combinaison de ces derniers, lors de l'exécution d'une requête complexe.

Si la technologie OLAP est le pur produit de l'entreposage de données, le forage de données est à un autre niveau et peut ne pas être associé à une base de données. On est passé des systèmes opérationnels aux systèmes stratégiques via les systèmes tactiques. Les systèmes opérationnels ou de production utilisent la gestion de données en transactionnel. Alors que les systèmes tactiques (OLAP) par l'entremise d'un infocentre, par exemple, aident grâce aux résultats qu'ils retournent à des prises de décisions rapides. Ces résultats sont généralement des courbes, des agrégats ou des histogrammes calculés aux moyens de somme, moyenne, maximum, minimum,...etc. Les systèmes stratégiques mettent à la disposition des décideurs des outils et des connaissances pour des choix d'orientations fondamentales selon les règles de fonctionnement de l'entreprise. Les événements passés sont décrits dans des modèles à partir desquels des évolutions futures peuvent être prédites.

Chapitre IV

Multi K-means: Une méta-méthode basée sur les K-means

1. DÉFINITION DES OBJECTIFS

L'exploration des données consiste en l'application de différents algorithmes de forage de données à un ensemble de données représentées sous la forme d'une table relationnelle. Le dit ensemble de données est souvent une sélection représentative de la population globale en ce qui concerne le problème à étudier et les objectifs à atteindre. Il existe différents types d'algorithmes de forage de données, et pour chaque type d'algorithmes une multitude de méthodes et de variantes. Le choix d'une ou de plusieurs méthodes dépend des objectifs qu'on veut atteindre, des types et de la qualité des données disponibles, de notre compétence à utiliser les outils de forage de données et des capacités des analystes et experts du domaine étudié à comprendre et interpréter les résultats obtenus.

Il est évident que toute population de patients, peut être fractionnée en un ensemble de catégories et cela selon plusieurs niveaux de granularité. La grande disponibilité des données historisées dans l'entrepôt de données CIRESSS nous donne cette opportunité d'essayer de trouver les différents sous-groupes que peut contenir une population de malades cardiaques. Pour cela nous avons œuvré comme suit :

1) Déterminer l'ensemble des données dont on aura besoin. Informations personnels, informations cliniques, informations biochimiques, ... et ce en se fiant aux experts du domaine et à l'expérience de l'équipe du CRED[5] tout en essayant de se dissocier le plus possible des idées pré-établies et des pratiques trop standards.

2) Extraction, nettoyage et transformation des données.

3) Détection et détermination des sous populations en appliquant certains algorithmes de forage de données. Pour ce faire, un choix judicieux des attributs à utiliser est indispensable. Une forte connaissance de la base de données et des attributs les plus significatifs pour notre étude de la part des experts du domaine nous évite d'aller vers des algorithmes de sélection d'attributs. Plusieurs méthodes seront expérimentées. Les différents résultats peuvent être ou bien complémentaires ou bien repris par une autre méthode à des fins d'agrégation ou de sélection.

4) Analyse des résultats obtenus et vérification si certains types de population qui nous intéressent ont été mis en évidence. Si c'est le cas nous continuerons à l'étape suivante sinon nous remettrons en cause les résultats et méthodes choisies en revenant à l'étape précédente.

5) La dernière partie du projet, intervient une fois que nos sous populations sont déterminées et les résultats approuvés. Elle consistera à faire un rapprochement entre la conduite pratique établie à partir de la littérature (articles, ...) dans le traitement des syndromes coronariens aigus et les pratiques routinières adoptées dans les deux hôpitaux CHUS et Hôtel Dieu de Sherbrooke. Pour cela une étude statistique sera sûrement indispensable, suivi d'une étude exploratoire si nécessaire. L'étude exploratoire sera basée sur les méthodes de classification pour, par exemple,

[5] CRED (Collaboration en Recherche pour l'Efficacité en Diagnostic) est un laboratoire du département de biochimie clinique du Centre Hospitalier Universitaire de Sherbrooke, avec lequel nous avons mené nos expériences.

déterminer le type de parcours médical que pourra suivre un patient et des méthodes d'estimation de certains paramètres pour, par exemple, pouvoir prédire la durée de séjour d'un patient à l'hôpital.

Du point de vue clinique, nos motivations portaient essentiellement sur les stratégies et associations d'utilisation des tests:

1) La stratégie d'utilisation des tests de laboratoire reste une question d'importance capitale relative à l'épineuse question du coût versus efficacité. L'abondance des tests en laboratoire qui ne cesse de croître et la diversité des conduites pratiques adoptées par les médecins, développées et proposées par les laboratoires, publiées dans les articles et revues font qu'on ne peut nous désintéresser de la complexité des approches que cela engendre et la diversité des stratégies qui peuvent en découler.

2) Exploration préliminaire de l'association d'utilisation des tests diagnostiques de biochimie important pour la cardiologie (troponine, enzymes cardiaques, ...) avec d'autres tests diagnostiques (tests cardiologiques, ...), procédures thérapeutiques (angioplastie), paramètres de gestion des soins (durée de séjour, ...) ainsi que les diagnostiques dans le but de découvrir:
   - Des patrons d'intérêts
   - Des sous populations de patients

3) Préparation du terrain pour des prochains projets de forage de données en association avec les experts des laboratoires et cliniques.

2. DÉFINITION DE LA POPULATION

Dans toute étude relevant de l'analyse d'un processus ou d'une population, il est primordial, après avoir bien défini les objectifs, de déterminer avec précision la population visée. Un tel choix est tributaire de plusieurs paramètres. Le premier est de vérifier si du point de vue qualitatif la population envisagée donne accès aux informations recherchées pour atteindre l'objectif visé. Deuxièmement, il est important de déterminer la taille nécessaire et suffisante de la population, en tenant compte des capacités de traitement et des coûts que cela peut engendrer. Il est aussi utile de vérifier qu'une population d'une telle taille est disponible et accessible. Le troisième point à considérer est d'envisager les techniques d'échantillonnage si la taille de la population est trop volumineuse. Plus la taille de la population est grande plus elle a de chances d'être représentative de la population globale. Il est donc plus intéressant de prendre une population la plus large possible quitte à procéder à un échantillonnage par la suite.

Notre étude devait porter sur les maladies cardiaques en général et sur les maladies coronariennes en particulier. Ce qui réduisait considérablement la taille de la population des patients en comparaison avec leur nombre total. Il nous restait à donner une définition précise du patient cardiaque. Nous avons relevé trois caractéristiques pouvant le décrire.

- *Un patient cardiaque est consulté par un cardiologue.*
- *Un patient cardiaque a au moins un diagnostic de type cardiaque.*
- *Un patient cardiaque a été admis au service de cardiologie.*

Ces trois caractéristiques bien qu'elles soient relatives à la cardiologie, ne sont pas spécifiques au patient cardiaque typique. Un patient peut être consulté par un cardiologue pour prévenir un problème de type cardiaque sans pour qu'il le soit forcément. Il peut aussi être admis au service de cardiologie comme il peut avoir un diagnostic secondaire de type cardiaque inhérent à ses antécédents alors qu'il consulte pour une toute autre maladie. Cette complexité de prise en charge des patients fait que la définition du cardiaque n'est pas évidente et prête souvent à confusion.

Un patient admis au service de cardiologie a généralement un problème cardiaque. Il est forcément consulté au moins une fois par un cardiologue et a souvent, au pire, un diagnostic secondaire de type cardiaque. Il regroupe donc les trois caractéristiques citées plus haut. C'est pourquoi nous avons décidé de limiter notre population de patients cardiaques à étudier aux patients admis au service de cardiologie. Un autre avantage d'un tel choix est la simplification de la recherche des patients. Le seul critère de sélection à préciser est l'admission au service de cardiologie.

Une fois la définition de la population éclaircie, il reste à déterminer sa taille. Une première contrainte à considérer est l'énorme quantité de données produite par la visite d'un patient. L'extraction des tests passés à quelques milliers de patients peut facilement atteindre le demi-million de lignes. La durée d'hospitalisation des patients cardiaques excède très rarement quelques semaines, même après une intervention chirurgicale. Les risques de rechute et de ré-hospitalisation sont grands les premiers jours et s'atténuent au fil des semaines. Rare sont les études qui parlent de suivi de plus d'un an. Sur la base de ces constatations, nous avons décidé de limiter notre étude à la seule année 2002. Ce choix nous permet de limiter la taille des données et est suffisant pour capturer la visite et le suivi d'un patient. Il peut être considéré comme un échantillonnage par groupe assez représentatif de la population cardiaque globale.

3. EXTRACTION DES DONNÉES

Définir la population de patients à considérer dans l'étude n'est pas suffisant. Il convient après de décider des données qu'il faut extraire. Ces derniers doivent pouvoir caractériser la population étudiée et offrir assez d'information pour répondre convenablement aux objectifs de l'étude. La nature sensible des données que nous avons à manipuler, les contraintes administratives par lesquelles nous sommes liés et surtout l'éthique de travail que nous nous sommes imposés, nous oblige à ne révéler des données que leurs descriptions et nous défendent dont révéler le contenu.

Pour une plus grande maîtrise des données, il nous a paru souhaitable de classer ces derniers selon leurs types. Nous avons pour cela relever six (06) classes de données.

3.1. Classes de données

Les données personnelles : représentent toutes les données identifiant un patient lors d'une visite. Elles contiennent le numéro de la visite qui est unique pour chaque visite et le numéro de patient qui identifie de façon unique chaque patient, le sexe, la race, l'âge à l'admission, ...

Les tests : Ils sont de 03 types. Les tests biochimiques, radiologiques et les signes vitaux. Il existe bien sûr d'autres types de tests qui ne sont pas spécifiques aux maladies cardiaques. La rareté de leur prescription dans le contexte de notre étude a fait qu'il n'est pas très pertinent de les considérer.

Les diagnostics : Ils sont de deux types, ceux considérés de type cardiaque et les autres, soit les non cardiaques. En première phase, tous les diagnostiques qu'ils soient principaux ou secondaires, cardiaques ou autres sont extraits de l'entrepôt de données. Seuls les diagnostics ayant une relation avec les maladies cardiaques seront considérés. Nous ferons une plus grande distinction à l'intérieur de ceux-ci par la suite.

Les traitements (Interventions et procédures) : Ils sont de deux types, ceux considérés de type cardiaque et ceux de type non cardiaque. Les interventions et procédures décrivent ce qui a été fait et non le résultat de l'opération. Pour avoir les résultats, il faut revenir aux diagnostics, aux tests et considérer le suivi du patient.

Les antécédents diagnostics : Ceux-ci sont tous les diagnostics établis pour un patient de la population considérée lors de ses visites précédentes. Si un patient a été admis 2 fois au service de cardiologie pendant l'année 2002, les deux visites seront sélectionnées pour notre étude et les diagnostics de la première visite seront considérés comme des antécédents diagnostics dans la deuxième visite. Comme les diagnostics, les antécédents diagnostics sont de deux types.

Les antécédents traitements : Ceux-ci sont tous les traitements établis pour un patient de la population considérée lors de ses visites précédentes. Ils sont de deux types, ceux considérés de type cardiaque et les autres, soit les non cardiaques.

3.2. Choix des données pertinentes

Pour parler de pertinence, il est utile d'associer aux données l'objectif assigné au projet. Même certaines données qui semblent pertinentes et nécessaires dans tout projet quelque soit son genre, se révèlent totalement inutiles voir nuisibles par le bruit qu'elles engendrent à l'étude. Notre travail portait sur les maladies cardiaques, et plus spécifiquement sur les syndromes coronariens aigus. Il s'inscrit dans un projet de modélisation des diagnostics cardiaques. Le choix des données devait se faire dans ce contexte selon les classes de variables définies plus haut.

a) Informations personnelles

Numéro du patient
Numéro de visite
Date de Naissance
Race
Statut Marital
Occupation

b) Tests biologiques et examens complémentaires

1) Tests biochimiques

Nom du test	Composante du test
Cholestérol HDL LDL Trigl	Cholestérol LDL, Cholestérol-LDL Cholestérol total Cholestérol-HDL Rapport C-LDL/C-HDL Rapport CT/C-HDL Triglycérides VLDL
Créatine kinase sérique	Créatine kinase
Créatine kinase totale et sa fraction MB, Profil enzyme cardiaque [CK]	CK, Créatine kinase CK-MB Renseignement Clinique
Créatinine sérique	Créatinine
Déhydrogénase lactique sérique	LD
Digoxin sérique	Digoxin
Electrolytes sériques [Na, K, Cl], Profil sérique de base [Na K créatinine]	Cl sérique, Créatinine sérique K sérique, Potassium sérique Na sérique, Sodium sérique
Glucose sérique	Glucose
Hémoglobine A1c	Hémoglobine A1c
Phosphates sériques	Phosphates sériques
Profil lipidique [Trig, Chol]	Cholestérol total Triglycérides
Thyrotropine sérique	TSH
Triglycérides sériques	Triglycérides
Troponine I	Troponine I
Troponine T	Troponine T
Urée sérique	Urée

2) *Tests cardiologiques*

Nom du test	Composante du test
ECG	Cardiologue Date dictée Endroit Examen fait Particular Raison
ECG effort	% FC max Arythmie Capacité max Cardiologue Durée effort FC 85% FC max atteinte FC max prédite FC pré-test Protocole Résul d'épreuve Segment ST
ECG effort pour MiBi	% FC max Arythmie Cardiologue Durée effort Segment ST
Echocardio trans-oesophagien	Contractilité Taille Val aort Val mitr Val pulm Val tric
Echocardiogramme adulte	Contractilité Mode M Val aort Val mitr Val pulm Val tric
Holter	Médicaments Raison

c) Tests radiologiques

Nom du test	Composante du test
Artériographie abdominale	Date dictée Facteurs de risques Raison Région concernée Rapport final
Artériographie aorte et membres inférieurs	Date dictée Facteurs de risques Raison Région concernée Rapport final
Artériographie thoracique	Date dictée Facteurs de risques Raison Région concernée Rapport final
Doppler artériel membre inférieur	Date dictée Charge(s) add. Raison Région concernée Rapport final
Doppler artériel membre supérieur	Date dictée Charge(s) add. Raison Région concernée Rapport final
Doppler carotide	Date dictée Charge(s) add. Raison Région concernée Rapport final
Doppler rénal	Date dictée Charge(s) add. Raison Région concernée Rapport final
Fluoroscopie pour pacemaker	Fluoroscopie temps Genre Raison Rapport final
Poumons	Date dictée Charge(s) add. Raison Région concernée Rapport final

3) Les soins infirmiers

Nom de l'opération	Composante de l'opération
Enseignement pacemaker	Pacemaker - Enseignement post-opératoire Pacemaker - Enseignement pré-opératoire Pacemaker - Enseignement prise du pouls
Evaluation admission - médecine/chirurgie	Allergie Taille Travail
Poids	Poids
Saturation O2	O2 Quantité O2 Type d'administration SO2
SV - TA Pouls Resp Temp	Pouls Pouls - description Respiration Respiration - description TA TA - description Température Température - voie

4) Urgence

Nom de l'opération	Composante de l'opération
HF/Régistre de la salle d'urgence	Civière Consultation Diagnostic Traumatisme

5) Monitoring cardiologique

Nom de l'opération	Composante de l'opération
Pacemaker - Installation/Changement	Indication UT
Pacemaker – Suivi	Amplitude auriculaire Amplitude ventriculaire Conduction rétrograde Fréquence maximale Fréquence minimale Hystérèse Implantation Pacemaker Mode Pace-stop Prise de Warfarine Reprogrammation Rythme sous-jacent Sensibilité auriculaire Sensibilité ventricul Test de seuil

d) Les diagnostics

Les diagnostics secondaires et principaux établis pendant la durée de visite du patient

Code CIM9
Description
Type

e) Les traitements

Les traitements secondaires et principaux administrés au patient pendant sa durée de visite

Code CIM9
Description

f) Les diagnostics antécédents

Tous les diagnostics secondaires et principaux établis lors des précédentes visites du patient s'il y a lieu

Code CIM9
Description
Type

g) Les traitements antécédents

Les traitements secondaires et principaux administrés au patient lors de ses précédentes visites s'il y a lieu

Code CIM9
Description

h) Les facteurs de risque

Antécédents familiaux
Athérosclérose
Coronarien connu
Diabète
Dyslipidémie
Hypertension artérielle
Sédentarité
Tabac

4. NETTOYAGE DES DONNÉES

S'il y a une étape qui reste peu couverte par la littérature et les travaux scientifiques c'est bien le nettoyage des données avant et après l'entreposage. La particularité d'une telle opération est qu'elle est difficilement automatisable. La multiplicité des circonstances et facteurs relatifs aux erreurs tels que leurs types, leurs diversités, les circonstances et les lieux de leurs apparitions, leurs prises en charge, les difficultés de leurs interprétations, le besoin des spécialistes du domaine pour leurs traitements efficaces, font que le traitement des erreurs qui est une opération simple d'apparence devient d'une complexité telle que sa prise en charge totale de façon automatique est presque impossible.

Le traitement des erreurs n'est qu'un aspect parmi d'autre dans le nettoyage des données. Avant de procéder au nettoyage il est essentiel de recenser tous les aspects qui y touchent et identifier les étapes de son exécution tout en gardant dans la mire les objectifs à atteindre. Nous avons identifié trois grandes classes de données à nettoyer. Les données personnelles, les diagnostics et traitements et finalement les données des tests.

4.1. Les diagnostics et les traitements

Les diagnostics et les traitements ont un statut particulier. Leur saisie dans le système informatique est du ressort d'archivistes spécialisés. Elle ne se fait pas mot par mot mais par le choix d'un code auquel est associé une description du diagnostic ou du traitement. Il en ressort que ces descriptions sont cohérentes et sans erreurs dans le contenu. Les seules erreurs peuvent provenir du choix du code fait par l'archiviste. Notre attitude était de considérer le choix des codes comme juste et relativement bien fait et les erreurs éventuelles comme un aspect à étudier et à faire ressortir par les analyses de datamining.

4.2. Les données personnelles

Elles sont de deux types, celles attribuées par le système informatique et celles inhérentes à chaque patient. Le numéro de patient et le numéro de visite sont attribués par le système informatique et sont, par conséquent, très fiables. Le numéro de visite est attribué au patient à chaque nouvelle visite, alors que le numéro de patient est unique pour chaque patient et reste inchangé. Le seul point à vérifier était la correspondance des numéros de patients entre les deux hôpitaux de *Fleurimont* et de l'*Hôtel Dieu*. Comme notre travail concernait l'année 2002 pendant laquelle ce problème avait déjà été réglé, les numéros de visite et les numéros de patient ne posaient aucun problème.

Pour les autres données personnelles, seules les données à saisie libre posaient problèmes. La date de naissance était correcte, il fallait donc traiter les redondances.

Les redondances peuvent être classées en 5 grands types :

1. *La redondance par la forme féminine*
2. *La redondance par la forme plurielle*
3. *La redondance par d'autres formes fléchies*[6]
4. *La redondance par les synonymes*
5. *La redondance due aux fautes d'orthographe.*

Le traitement de tels types de redondances se fait grâce à la construction d'un dictionnaire en trois étapes:

[6] Forme fléchie : autre forme que peut prendre un mot. Le traitement de la forme fléchie se fait par la lemmatisation qui consiste à ramener le mot à sa forme standardisée, soit l'infinitif, le singulier et la racine.

1. *Construction initiale d'un dictionnaire de données, de façon manuelle à partir d'une liste ordonnée des données extraites à partir de CIRESSS.*
2. *Étiquetage d'un nouveau mot rencontré et n'appartenant pas au dictionnaire comme candidat potentiel à ajouter au dictionnaire comme nom ou comme une nouvelle forme d'un nom existant.*
3. *Mise à jour manuelle du dictionnaire selon l'appréciation de l'utilisateur, à partir des mots précédemment étiquetés.*

Le dictionnaire est composé de trois tables. Celle des mots fléchis qui contient le mot, sa forme féminine, sa forme pluriel et toutes les autres formes rencontrées dans les données extraites et enrichi par d'autres proposées par l'utilisateur. La table des synonymes qui contient le mot et tous ses synonymes recensés. La table des fautes d'orthographe potentielles qui contient le mot et les erreurs communes relatives au mot.

Ce type de traitements a été appliqué à l'occupation du patient et à une moindre mesure, à la race. L'adresse n'étant pas disponible n'a pas été traitée.

4.3. Les tests

Le traitement des valeurs de tests n'est pas chose évidente. La diversité des appareils de mesures et l'intervention de l'être humain font qu'il peut exister différents formats de données pour un même type de test. Le nettoyage de telles données est de 6 types :

4.3.1. Identification des formats

Toutes les données extraites de CIRESSS sont de type texte, ce qui rend difficile l'identification du format du test. Un premier travail est d'associer à chaque test son format le plus approprié. L'identification automatique est possible mais reste difficile et aléatoire. L'identification d'un format par la fréquence ne garantie pas le bon résultat. Par exemple, la valeur du test CK-MB est de type numérique avec 2 chiffres décimaux, alors que plus de 90% de ses valeurs sont de type alphabétique. La raison est que le test n'est pas effectué dans la plupart des cas, parce qu'il n'est pas pertinent pour le diagnostic suspecté, d'où le remplacement de sa valeur par une remarque textuelle.

Pour une bonne identification des formats, nous avons créé un dictionnaire des tests qui recense tous les tests que nous avons extraits à partir de CIRESSS. À chaque test est associé un format selon ses valeurs. La validation par les experts du domaine était nécessaire. La construction du dictionnaire des tests ne concernait que les tests choisis pour le projet. Pour une généralisation du processus, il serait utile dans une phase ultérieure d'extraire tous les tests présents dans l'entrepôt de données CIRESSS et de construire un dictionnaire plus complet. Le travail est fastidieux mais tout à fait réalisable dans des délais raisonnables.

Le traitement de tels types de redondances se fait grâce à la construction d'un dictionnaire en trois étapes:

1. *Construction initiale d'un dictionnaire des tests, de façon manuelle à partir d'une liste ordonnée des tests extraits à partir de CIRESSS.*
2. *Proposition de format à un nouveau test rencontré et n'appartenant pas au dictionnaire.*
3. *Mise à jour du dictionnaire selon le format choisis par l'utilisateur.*

Les formats sont de quatre types, les entiers, les décimaux avec un nombre fini de chiffres avant et après la virgule *(ex: 999,99)*, les valeurs alphanumériques contenant les valeurs textuelles et les remarques et quatrièmement les dates. Le type date ne concerne pas la valeur du test mais peut lui être associé pour préciser la date et l'heure exacte du test.

4.3.2. Traitement des erreurs

Nous avons relevé certaines valeurs de test qui nous ont paru anormales. Une valeur trop grande ou trop petite est souvent révélatrice d'une erreur de saisie. Comme nous ne voulions pas faire de spéculation sur la nature et l'origine des erreurs, nous avons adopté une politique claire :

1. *Si une valeur de test appartient au domaine du possible, même si elle est considérée comme anormale dans le contexte où elle a été recueillie, elle est prise telle qu'elle.*
2. *Si aucune signification ou interprétation ne peut être attribuée à une valeur de test, elle est tout simplement ignorée et est considérée comme non disponible.*

4.3.3. Suppression des unités de mesure

Le serveur Nucleus de l'entrepôt de données CIRESSS retourne les valeurs de tests concaténées à leurs unités de mesure, si elles existent. Il est donc primordial de dissocier les unités des valeurs mesurées. Une première étape à réaliser est d'identifier le format des données mesurées, chose qui a été traitée dans le titre précédent. Il faut aussi recenser toutes les unités en association avec les tests, pour pouvoir les utiliser ultérieurement dans les conversions possibles ou pour l'affichage des valeurs de tests. La création d'une liste d'unités s'est alors imposée d'elle même. Elle se fait en deux étapes :

1. *Construction initiale de la liste des unités de tests, de façon manuelle à partir d'une liste ordonnée des valeurs de tests extraites à partir de CIRESSS.*
2. *Association de chaque test à son unité correspondante si elle existe.*

Une fois les unités recensées et les associations entre les tests et leurs unités établies, il est possible de supprimer les unités des valeurs extraites. Dans notre travail, nous n'avons traité que les tests qui nous intéressaient. Le recensement de toutes les unités de mesure présentes dans l'entrepôt de données CIRESSS et leurs associations avec les tests correspondant est une autre tâche à compléter.

4.3.4. Traitement des dates

Il existe deux types de date dans CIRESSS. Les dates introduites par le concepteur de l'entrepôt de données afin de gérer la dimension du temps relative à toute conception multidimensionnelle et les dates importées à partir du système de production et qui sont associées à chaque saisie de données. Si pour les premières il n'existe aucun problème, elles sont même utilisées comme clés primaires et étrangères pour la gestion du temps, les deuxièmes présentent de grave lacunes. Leurs formats de données ne sont pas standardisés. Il était alors primordial de recenser tous les formats possibles et toutes les erreurs rencontrées pour les traiter et les transformer en un seul format standard. Pour les dates complètes et correctes, il suffisait de les convertir au format choisi. Le traitement se fait comme suit :

1. *Choix d'un format standard : yyyy/mm/jj HH:MM*
2. *Identification de tous les formats de date rencontrés dans CIRESSS*
3. *Traitement des erreurs*
4. *Écriture des fonctions de transformation des formats rencontrés en format standard*

Le traitement des erreurs est de deux genres :

1. *Date sans année : Attribution de l'année de la visite si il n'y a pas d'ambiguïté*
2. *Date sans jour : Estimation du jour du test et correction de la date en conséquence si il n'y a pas d'ambiguïté.*

L'ambiguïté est due au chevauchement de la date de la visite sur deux années ou deux jours successifs. La résolution d'une telle ambiguïté nécessite une investigation plus poussée pour estimer la date exacte, si c'est possible, en considérant tous les autres éléments de la visite.

4.3.5. Traitement des remarques

Une remarque est une note médicale affectée à un test qui est de type numérique. La remarque peut être due à plusieurs raisons. Le plus souvent, la valeur du test est hors des limites de lecture du marqueur[7] utilisé, les méthodes de lecture ne pouvant capturer la valeur exacte.

4.3.6. Traitement des valeurs manquantes

Le traitement des valeurs manquantes est un véritable casse-tête auquel font face tous les utilisateurs. Le nombre de tests qui ont été fait mais dont la valeur n'a pas été introduite dans le système informatique est très restreint, voir même négligeable. Néanmoins, il fallait traiter le peu de cas rencontrés. Une seule solution a été adoptée. Ignorer les tests ayant ce genre d'erreur et les considérer comme non faits.

Il existe un deuxième type de valeur manquante, beaucoup plus subtile et difficile à interpréter. Ce sont les tests qui, en théorie, devaient être faits mais qui ne l'ont pas été. Comme le test n'a même pas été commandé, on ne peut considérer sa valeur comme manquante. À chaque début de phase d'analyse, une

[7] Substance détectable et dosable par des méthodes de biochimie et correspondant à la présence ou au développement d'une pathologie.

valeur, désignant l'absence du test, doit lui être attribuée. La solution était de différer le traitement des valeurs manquantes de tests au début de la phase d'analyse, pour au moins une raison majeure.

> *Une valeur manquante est traitée différemment selon le type de test et de l'étude qu'on veut en faire.*

La valeur du pouls d'un patient à antécédents coronariens se présentant avec des douleurs thoraciques n'est pas la même que l'absence du test de la troponine[8] qui reste quasi-systématique pour un tel patient. Aussi, un ECG manquant pour un malade cardiaque qui arrive à l'urgence sans douleurs persistantes et un autre se présentant avec un infarctus de myocarde ne peut pas être considéré comme une même chose. Pour le premier, l'absence d'ECG[9] est possible, pour le deuxième l'ECG est indispensable. Les raisons de son absence doivent être mises en évidence.

Il est alors nécessaire, voir indispensable de différer le traitement de ce genre de valeurs manquantes au début de la phase d'analyse, plutôt que d'en faire un même et unique traitement valable pour toutes les études.

5. TRANSFORMATION DES DONNÉES

Une fois le nettoyage effectué, les erreurs, les formats de données, les valeurs manquantes, les valeurs erronées,... sont traitées pour pourvoir exploiter les données de façon correcte et aussi minimiser la marge d'erreur des résultats. Mais, le nettoyage à lui seul n'est pas suffisant comme pré-traitement des données. Les spécificités des méthodes de datamining en terme de format, volume et type de données d'entrée, la qualité des données, leur volume total, leurs types et les résultats souhaités et les objectifs à atteindre peuvent exiger d'autres traitements de transformation. Nous avons, donc, procédé à deux types de transformation, le regroupement des diagnostics et des traitements par type et la discrétisation de certains tests tels que la Troponine et l'ECG.

5.1. Regroupement des données par type

Le but principal du regroupement des données par type est d'éviter de noyer les résultats dans leurs détails et leurs particularités. Pour que les méthodes de classifications soient efficaces et leurs résultats intéressants et pour augmenter les chances de généralisation, il est recommandé d'éviter de considérer trop de détails, surtout si le nombre d'individus vérifiant ces détails est petit ou insignifiant. C'est pourquoi nous avons décidé de regrouper les diagnostics et les traitements par type.

---

[8] La troponine est un complexe de protéines dont les sous-types I et T sont utilisés pour diagnostiquer chez les patients atteints de douleurs thoraciques, l'angine de poitrine instable et l'infarctus du myocarde.

[9] ECG ou Électrocardiogramme est un test diagnostic qui évalue l'activité électrique du cœur.

5.1.1. Classification des diagnostics en catégories

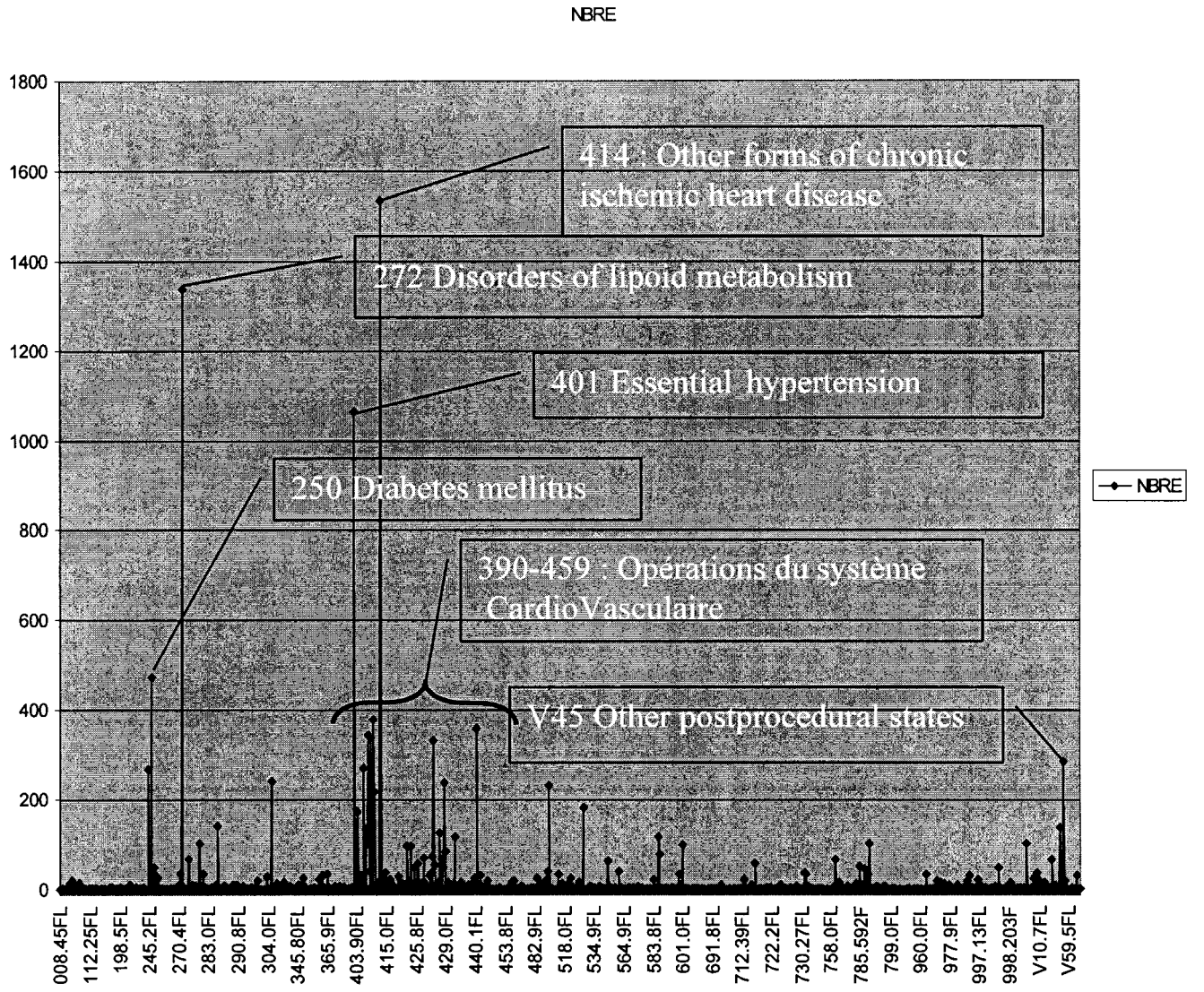

Figure 16 : Fréquence des diagnostics

Les diagnostics sont déjà classés en groupes ou chapitre selon le code CIM, classification internationale des maladies. Il s'agit d'une classification monoaxiale composée de 21 chapitres dont 17 concernent des maladies et quatre concernent les signes et résultats anormaux, les causes de traumatismes, d'empoisonnement ou de morbidité, l'état de santé et les facteurs de recours aux soins. Les catégories de maladies sont définies en fonction d'un caractère commun qui peut être l'étiologie (1 = Maladies infectieuses, lettres A et B), la topographie (9 = maladies de l'appareil circulatoire, lettre I), la physiologie (15 =Grossesse et accouchement, lettre O) ou la pathologie (II = Tumeurs). La classification aboutit par subdivisions successives à un code à 3 caractères (une lettre correspondant au chapitre puis 2 chiffres) pour les maladies définies à un niveau général, décliné par l'ajout d'un quatrième chiffre (après un point) pour désigner les diagnostics précis et les formes cliniques; le sous-code 9 désignant l'absence de précision (SAI = sans autre indication) et le sous-code 8 les autres formes non précédemment définies. Dans certains cas, un cinquième chiffre a été rajouté afin d'améliorer la finesse de la description. Le service des archives de l'hôpital a ajouté d'autres sous-codes locaux, non définis dans le CIM9.

Le CIM9 est représenté sous la forme d'un seul arbre hiérarchique dans lequel une entité pathologique est représentée une seule fois dans la classification, ce qui peut poser des difficultés. Ainsi toutes les tumeurs sont regroupées dans un même chapitre et ne sont pas dans leur chapitre d'appareil. Il arrive, aussi, qu'une même maladie apparaisse en deux endroits distincts (avec deux codes). Ces contraintes de représentation nous ont dissuadé de choisir les chapitres et les sous-chapitres de la classification CIM9 comme catégories.

Les critères retenus pour le regroupement des diagnostics sont d'une part leurs similarités du point de vue médical et leurs fréquences d'apparitions. La similarité a été défini par le médecin participant au projet en s'appuyant sur le tableau des fréquences résumé par la *figure 16*. Néanmoins, le code CIM, adopté par l'hôpital nous a été d'une grande utilité. Les catégories choisies sont représentées dans le *tableau 11*.

code	Nom	Diagnostics
01	Hypercholestérolémie	Hypercholestérolémie
02	Valvulaire	Valvulaire
03	Autre cardiopathie non valvulaire	Autre cardiopathie non valvulaire
04	Hypertension Artérielle	Hypertension artérielle
05	Maladies coronariennes	Ischémie Athérosclérose Infarctus Angine Sténose coronarienne MCAS (Maladie Coronarienne Athérosclérotique Stable) Embolie

06	Inflammation (ite)	Inflammation (ite)
07	Anévrisme du cœur	Anévrisme du cœur Abcès du cœur
08	Cœur pulmonaire	Cœur pulmonaire
09	Myocardiopathie	Myocardiopathie
10	Trouble du Rythme	Trouble du Rythme
11	IC, OAP	IC, OAP (Insuffisance Cardiaque, œdème aigu pulmonaire, ...)
12	Terminologie Générale	Terminologie Générale MCV, Vasculaire, Choc, Collapsus, Syncope, ...
13	Transplantation cardiaque	Transplantation cardiaque
14	Anomalie du cœur	Anomalie du cœur (Congénitale ou autre)
15	Stimulateur	Stimulateur
16	Diabète	Diabète
17	Pathologie cérébrale	Pathologie cérébrale
18	Pathologie rénale	Pathologie rénale
19	Intoxication médicamenteuse	Intoxication médicamenteuse
20	Complication post-chirurgicale	Complication post-chirurgicale
21	Arrêt cardiaque	Arrêt cardiaque
22	Maladie veineuse des extrémités	Maladie veineuse des extrémités (Phlébite)

Tableau 11 : Catégories de diagnostics

5.1.2. Classification des traitements en catégories

Les critères retenus pour le regroupement des traitements en catégories sont pratiquement les mêmes que ceux choisis pour les diagnostics. Il s'agit de leurs similarités du point de vue médical et leurs fréquences d'apparitions. Comme les diagnostics, les traitements ont eux aussi un code CIM, affecté par les archivistes de l'hôpital. Codes qui ont été d'une grande utilité pour le regroupement des traitements. La *figure 17* résume le tableau des fréquences, alors que le *tableau 12* résume les catégories de traitements retenus.

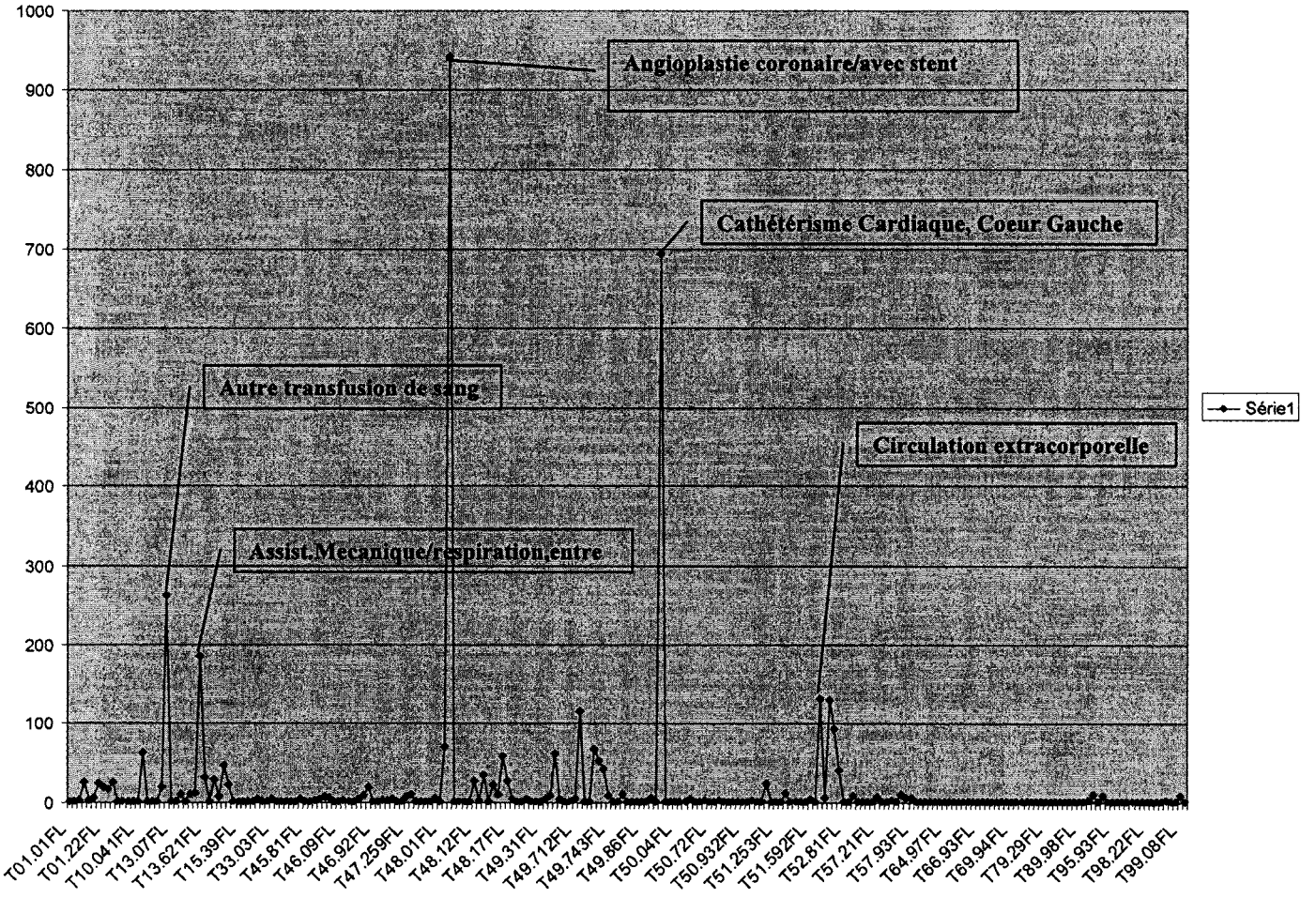

Figure 17 : Fréquence des traitements

code	Nom	Traitements
1	Examen électrique	écho cardiaque (trans-oesophagien,...)
2	ElectrochocMassage	scintigraphie/etude du fonct.cardiovasc./hematopoiet. injection iode 131
3	Vulve mitrale	cardioversion électrique cardioversion par overdrive pacing implantation de pacemaker implant.d'électr.myocardiques, implant.d'électr.endocardiques, remplacement d'électrodes endocardiques révision ou replacement d'électrodes remplacement générateur pulsatif remplacement de pile ablation d'électrodes myocardiques ablation d'électrodes endocardiques ablation du système de pacemaker, sans remplacement
33	vulve aortique	
4	Terme général	défibrillation autre électrochoc cardiaque massage cardiaque a thorax ferme autre modification du rythme cardiaque
5	Artère coronaire	valvuloplastie a coeur ouv.sans rempl.,valv.aortique remplacement valvule mitrale/st-jude remplacement/révision de prothèse mitrale autre remplacement de la valvule mitrale remplacement de la valvule aortique avec greffe tissulaire remplacement valvule aortique annuloplastie autre opération sur autre structure contiguë aux valves cardiaques valvuloplastie percutanée
6	Angioplastie	répar.de communication interventric. avec greffe tiss.
7	Opération vaisseaux	désobstruction d'artère coronaire angioplastie coronaire sans et avec stent angioplastie coronaire+thrombolyse sans et avec stent angioplastie coronaire à thorax ouvert implantation de ballon pulsatif ablation ballon intra-aortique cathétérisme cardiaque, coeur droit ou gauche catherinette combiné du coeur droit et du coeur

		gauche angioplastie avec insertion de prothèse artérielle angioplastie per-cutanée par ballonet
8	Drainage	pontage aortocoronaire pour revascularisation pac d'une artère coronaire avec saphène/allogène pontage aortocoronaire d'une artère coronaire pontage x2 aorto-coronarien avec saphène pontage aortocoronaire de deux artères coronaires pontage x3 aorto-coronarien avec saphène reprise de pac de 3 artères coronaires pontage aortocoronaire de trois artères coronaires pontage x4 aorto-coronarien avec saphène pontage aortocoronaire de quatre artères coron. ou+ pontage unique ou double entre artère mammaire et artère coron. pontage x6 aorto-coronarien autre pontage pr revascularisation cardiaque incision de l'aorte résection de l'aorte avec anastomose résection de l'aorte avec remplacement implantation d'un umbrella plicature de la veine cave / parapluie opération de bentall pontage aorto-iliaque pontage aorto-femoral bypass (pontage) aorto-iliaque-femoral pontage aorto-mésentérique supérieure pontage aorto-aortique
9	Réparation cœur	artériographie coronaire utilisant un cathéter unique
10	Ballon	autre opération sur les vaisseaux du coeur
11	Puce électrique	péricardiocentèse drainage péricarde fenestration du péricarde biopsie du coeur
12	Cathétérisme	Péricardiotomie exérèse d'anévrisme du coeur ablation par cathéter de lésions du coeur (fulguration) exérèse d'autres lésions du coeur (sans cathéter) réparation du coeur et du péricarde
13	Aorte	ablation de système d'assistance cardiaque assistance mécanique/respiration thoracotomie / drainage extrapleural réouverture thoraco.récente/arrêt d'hémorragie réouverture de siège de thoracotomie récente suture d'une artère suture d'une veine profondoplastie
14	Parapluie	circulation extracorporelle cardioplégie

		pacemaker cardiaque durant une intervention autre transfusion de sang
15	Transfusion sanguine	
16	Assistance mécanique	
17	Inconnu	

Tableau 12 : Catégories de traitements

5.2. Discrétisation

L'intérêt de la discrétisation est, d'une part, la récupération des seuils des différents tests établis par la pratique. Seuils qui sont le fruit d'une synthèse des valeurs établies par les producteurs des produits utilisés dans les tests, les différentes études qui traitent des tests et la pratique dans l'hôpital. D'autre part, la discrétisation permet de simplifier le traitement des données en passant d'un nombre de quelques centaines de valeurs à moins d'une dizaine. Un autre avantage de la discrétisation est qu'il facilite grandement l'interprétation des résultats et permet de reproduire les repères sur lesquels les experts du domaine se basent pour leurs analyses.

La méthode de discrétisation choisie est de type dirigé, les intervalles de discrétisation ne sont pas égaux, mais obéissent à la logique scientifique établi par la pratique. L'inconvénient de la discrétisation est qu'elle peut inhiber certains détails de l'attribut qu'elle traite. Cet inconvénient qui peut être handicapant dans certains cas, reste acceptable dans notre travail, vu la simplification que la discrétisation apporte à la complexité de la structure des données étudiées.

5.2.1. Discrétisation de l'ECG :

L'ECG nous a posé un réel problème. Il est sauvegardé sous sa forme la plus brute. C'est un rapport de plusieurs lignes que le médecin rédige après chaque examen d'ECG. Le rapport ne respecte aucune norme et chaque médecin y va de son propre style. Vu la complexité de traitement de tels rapports, nous avons opté pour une discrétisation radicale qui consiste à ne considérer que la présence ou l'absence de l'ECG. La présence ou non d'un ECG au début de l'hospitalisation d'un patient présumé cardiaque, bien qu'elle ne soit pas très révélatrice, reste assez discriminante quant à l'état de santé du malade.

Une autre solution que nous avons entamé et que nous n'avons pas pu finir et de traiter le contenu des rapports pour en extraire l'information pertinente. L'idée est d'identifier tous les termes utilisés par les médecins. Nous avons recensé quatre types de termes reliés à un organe, l'organe lui même, une partie ou un endroit d'organe, sa description et finalement son comportement. Une fois les termes identifiés, il suffit de recenser les associations de certains termes pour pouvoir en déduire l'analyse du médecin. Une telle proposition reste faisable à cause du volume réduit du domaine des termes utilisés par les médecins et à l'évidence des relations entre les termes.

Exemple :

1. Rythme sinusal à 55/min. Nécrose inférieure et postérieure ancienne.

Termes retenus :	organe	: cœur	
	Endroit	: inférieure, postérieure	
	Description	: nécrose	
	Comportement	: rythme sinusal 55/min	
Associations :	Le cœur	***a***	un rythme sinusal 55/min
	Le cœur	***a***	une nécrose
	La nécrose	***est***	inférieure
	La nécrose	***est***	postérieure

2. Atteinte de la paroi postérieure.

Termes retenus :	organe	: cœur	
	Endroit	: paroi postérieure	
	Description	: Atteinte	
	Comportement	: aucun	
Associations :	La paroi postérieure	***est***	atteinte

La construction d'un dictionnaire des termes et d'une table d'associations possibles permet de faire un résumé du rapport de l'ECG du médecin d'une façon automatique. Il reste à traiter certaines ambiguïtés. Il arrive, par exemple, que des termes comme «*possible*», «*possiblement*», «*ancien*», «*récent*» accompagnent une description ou qu'un qualificatif d'un terme qui est, habituellement, au début de la phrase soit mis à la fin de la phrase. Associer chaque qualificatif au bon terme qu'il qualifie, est parfois pas du tout évident. Les techniques de textmining et de traitement du langage naturelle peuvent être explorées pour tenter de résoudre le problème.

5.2.2. Discrétisation de la troponine:

Il existe deux types de troponine, la «*I*» notée «*TnI*» et la «*T*» notée «*TnT*». Notre étude concernait l'année 2002, période à laquelle l'hôpital est passé de l'utilisation de la TnI à la TnT. Comme la sensibilité des deux troponines est légèrement différente et les seuils d'interprétation ne sont pas identiques, nous avons établi, pour chacune des troponines, cinq intervalles numérotés de 1 à 5. Nous avons, par la suite, fait la supposition que les cinq intervalles des deux troponines sont sensiblement équivalents.

Intervalle	TnI	TnT	Description
01	TnI < 0,1	TnT < 0,03	Troponine négative
02	0,1 ≤ TnI < 1,5	0,03 ≤ TnT < 0,10	Suggestif de dommage myocardique mineur
03	1,5 ≤ TnI < 3,5	0,10 ≤ TnT < 0,30	Suggestif d'infarctus du myocarde niveau I
04	3,5 ≤ TnI < 10	0,30 ≤ TnT < 1,10	Suggestif d'infarctus du myocarde niveau II
05	TnI ≥ 10	TnT ≥ 1,10	Suggestif d'infarctus du myocarde niveau III

Tableau 13 : Discrétisation de la troponine

Les intervalles ont été établis selon les seuils utilisés dans l'hôpital et la répartition des malades en fonction du diagnostic principal.

6. EXPLORATION DES DONNÉES

Comme expliqué au chapitre II, l'exploration des données peut prendre plusieurs formes. Cela dépend fortement des données à traiter et des objectifs à atteindre. L'exploration englobe la classification non supervisée, la génération des règles associatives, la classification supervisée, l'estimation et la prédiction de paramètres. Si les trois derniers types de techniques supposent l'existence de classe prédéterminées ou une connaissance forte du contenu de la base de données, les deux premières font de l'exploration primaire qui consiste à découvrir des patrons de données possiblement inconnus. Comme notre travail est une première partie d'un projet plus global, il était évident que nous devions commencer par ce type d'algorithmes de datamining afin de mieux connaître notre base de données et essayer d'en extraire le plus de connaissances possible qui puissent nous aider à mieux comprendre les données et nous défrichent le terrain pour nos études futures. Nous avons délaissé, dès le début de l'étude, les règles associatives, à cause du grand volume de données dont on disposait, et par conséquent, du nombre de règles que peut engendrer une génération directe de règles sur la base de données initiale. Nous avons, donc, opté pour la classification non supervisée et sur les k-means en particulier.

6.1. Choix des attributs

Le nombre initial d'attributs était de 130, que nous avons choisi parmi quelques milliers d'attributs constituant l'entrepôt de données. Nous avons déterminé cinq types.

Information personnelle : N° du patient, N° de la visite, l'âge, le sexe, la race, l'occupation, ...

Prise en charge : Durée de séjour

Les tests subis : Le nombre de tests qu'un malade cardiaque peut subir et leur diversité, nous a contraint de regrouper les tests selon différents types. Cette catégorisation a été pensée dans le but d'unifier les approches de traitement des tests de même type. Par exemple, dans le cas des enzymes et des

marqueurs, la valeur du premier test et la valeur maximale du même test qu'un malade subi sont les valeurs les plus intéressantes. Aussi, le moment où a été effectué le test par rapport à l'arrivée du malade à l'hôpital et par rapport aux premières douleurs est très important. Ces deux caractéristiques plaidaient pour le regroupement des enzymes et marqueurs dans une même catégorie. Le *tableau 14* résume les quatre types de tests que nous avons défini.

Électrolyte/Rénal	Enzymes / Marqueurs	Lipides	Autre test
Na : Sodium	CK : Créatine Kinase	Cht total	Glucose
K : Calcium	CKMB : Fraction MB	HDL	TSH
Cl : Chlore	TnI : Troponine I	...	Promplets
Urée	TnT Troponine T		...
Créatinine, ...	...		

Tableaux 14 : Regroupement des tests par catégories

Les diagnostics : Nous avons considérés tous les diagnostics qui ont été établis lors de la visite du malade. En effet, un malade peut avoir plus d'un diagnostic, dont un seul est considéré comme principal et les autres comme secondaires.

Les diagnostics antécédents: Nous avons pris la décision de considérer, s'il y a lieu, tous les diagnostics qui ont été établis lors des précédentes visites du malade. En effet, nous avons estimé qu'un malade qui a déjà séjourné à l'hôpital et pour lequel un diagnostic a été établi, ne peut être considéré comme un autre malade qui n'a jamais séjourné à l'hôpital ou pour lequel aucun diagnostic n'a été établi. Un diagnostic antécédent révèle deux informations. La première est l'hospitalisation antérieure du malade et la deuxième est le diagnostic lui même.

Les traitements: Par traitements, nous signifions les procédures et les interventions que le malade a subies pendant toute la durée de son hospitalisation, qu'elles soient spécifiques aux causes de l'admission ou non, et qu'elles correspondent au diagnostic principal ou pas.

Les traitements antécédents: Comme pour les diagnostics, nous avons préféré considérer tous les traitements que le malade a subi lors de ses éventuels précédents séjours à l'hôpital. En effet, un traitement antérieur peut révéler des informations pertinentes pouvant expliquer l'état actuel du patient.

6.2. Stratégie de choix des attributs

Afin de ne pas perdre trop d'informations et pour une utilisation optimale du contenu des données extraites, nous voulions garder toutes les données qui étaient à notre disposition. Mais, la complexité de prise de décision et d'établissement d'un diagnostic, le nombre infini de cheminements possibles qu'un patient se présentant à l'hôpital peut suivre, la différence d'approche entre les médecins, ont fait qu'il était quasiment impossible qu'un seul modèle informatique puisse représenter de façon correcte et fidèle toute

la base de données. Nos convictions se sont vite vérifiées quand on a essayé différents algorithmes de datamining sur les données extraites. Les modèles générés ne pouvaient être interprétés d'aucune façon. Les détails que contenaient les données faisaient qu'il y avait une trop grande diversité qui ne pouvait être capturée par aucun modèle.

Nous pouvons expliquer l'inefficacité des algorithmes de datamining sur les données extraites par :

1. la complexité de prise de décision et d'établissement d'un diagnostic.
2. Les détails non capturés par le système informatique mais essentiel pour le médecin.
3. Le nombre important de cheminements possibles qu'un patient se présentant à l'hôpital peut emprunter.
4. la différence d'approche entre les médecins.
5. L'intégration de nouveaux tests et possible changement d'interprétation des anciens.
6. Le changement dans la stratégie d'approche avec l'adoption par les médecins de l'hôpital d'un nouvel organigramme.

La solution adoptée est de procéder par objectif précis et par débroussaillage. Nous nous sommes alors concentrés sur la modélisation des patients cardiaques, ayant comme diagnostic final une des *maladies coronariennes* que sont les *cardiopathies ischémiques (CPI)*, les *infarctus, les* maladies *cardiaques athérosclérotique (MCAS), les angines instables* ainsi que d'autres diagnostics coronariens moins fréquents. Ceux-ci sont des diagnostics cardiaques typiques, fréquents et très proches les uns des autres. Pour ce faire, nous devions sélectionner les attributs les plus appropriés et avec lesquels, il est possible de générer un modèle le plus complet possible.

Nous avons repris les huit types de données que nous avions établis au début *(tableau15)*.

Personnel	Prise en charge	Tests	ECG	Antécédent diagnostics	Antécédent traitement	Diagnostics	Traitements
N°patient N°Visite Âge sexe	Durée de séjour	CK-MB : 1ere CK-MB : Max Troponine I Troponine T	ECG	Hypercholestérolémie Hypertension Artérielle Diabète Arrêt Cardiaque Autre Facteurs de risque Maladies Coronariennes	Procédure de Dilatation Chirurgie CardioVasculaire Médicament	Maladies Coronariennes CPI INF MCAS AI Arrêt Cardiaque	Procédure de Dilatation Chirurgie CardioVasculaire Artériographie Coron.

Tableau 15 : Liste des attributs sélectionnés

6.3. Classification non supervisée par le clustering

Comme expliqué précédemment, nous devions faire une exploration non supervisée des données afin de découvrir des patrons de données possiblement inconnus. Le choix des K-means s'est imposé de lui même pour différentes raisons. Le volume des données et son exploration primaire, sans but spécifique et précis nous imposaient une méthode simple et rapide. Le nombre d'attributs et le nombre de fois où la méthode devait être expérimentée ne nous permettaient pas d'opter pour des méthodes à forte notion statistique qui avait des temps d'exécution largement supérieur à celui du K-means. Nous avons testé des algorithmes tels que EM et Cobweb. Les résultats préliminaires qu'ils ont générés et le temps d'exécution nous ont découragé. Même si notre maîtrise des paramètres de telles méthodes restait limitée, il était clair que de telles méthodes étaient plus appropriées dans le cas d'objectif précis avec un nombre de groupes connus afin de paramétrer les méthodes le mieux possible, pour aboutir aux résultats souhaitées. Pareils détails, combinés à notre niveau de compréhension des données, rendaient l'exploitation de la méthode difficile et aléatoire. D'autres parts, les K-means sont très rapides en terme de temps d'exécution et génèrent des résultats beaucoup plus cohérents et plus simples à interpréter.

6.3.1. Détermination de la fonction de distance

Il existe plusieurs fonctions de distance qui peuvent être utilisées pour comparer les objets dans la méthode des K-means. Notre choix s'est porté sur la fonction la plus simple et la plus utilisée dans le domaine de la recherche et des applications. La distance euclidienne a de nombreux avantages par rapport à toutes ses rivales. Elle reste simple à calculer et permet de diminuer l'effet des valeurs extrêmes qu'elles soient erronées ou non. Nous avons rencontré et traité trois types de cas de valeurs.

Valeurs numériques : Elles sont normalisées, c'est à dire qu'elles sont ramenées à des valeurs entre 0 et 1. Le 1 représentant la valeur maximale que peut prendre l'attribut. La normalisation est indispensable puisque les unités de mesures des valeurs numériques ne sont pas les mêmes. Aussi, ces mêmes valeurs peuvent prendre plus de poids que les autres types de valeurs si elles ne sont pas normalisées. Les tests sont des exemples de valeurs numériques.

Objets d'une liste : Les éléments d'une liste finie sont comparés entre eux. Si deux individus ont la même valeur pour un attribut donné, la distance entre les deux individus pour cet attribut est égale à zéro, sinon, elle est égale à un. Les diagnostics et les traitements sont des exemples d'objets de liste.

Valeurs binaires : Un attribut peut prendre deux et seulement deux valeurs. Si deux individus ont la même valeur pour l'attribut binaire considéré, la distance entre les deux individus pour cet attribut est égale à zéro, sinon, elle est égale à un. La valeur binaire est un type particulier d'objet d'une liste. Le sexe et la présence ou l'absence de l'ECG sont des exemples de valeur binaire.

6.3.2. Capture de la notion de temps

Le choix de la fonction de distance est généralement suffisant pour utiliser la méthode des K-means. Notre préoccupation était d'aboutir à un modèle qui représenterait le plus fidèlement possible les données que nous avions à traiter. Hors, nous savons tous que l'état de santé d'un patient à l'instant τ dépend des paramètres recueillis ou mesurés à cet instant (questionnaire, tension artérielle, température, douleurs,...) mais aussi, d'autres paramètres issus de son dossier, s'ils existent. Un médecin, qui consulte le dossier de son patient, tient toujours compte des dates où l'information a été ajoutée. Un cardiaque qui n'a été diagnostiqué comme hypertendu que le jour de la visite, ne peut pas être considéré de la même façon qu'un cardiaque qui a un problème d'hypertension depuis dix ans. Hors, le suivi dans le temps d'un individu est très mal capturé par les méthodes de datamining. Il n'y a que les règles associatives qui peuvent le faire, mais elles restent limitées par le nombre de règles générées. Plus, on ajoute d'événements temporels plus grand est le nombre de règles qui en découlent.

L'idée est de pondérer la valeur d'un paramètre donné en fonction de la date où elle a été recueillie. Dépendamment de la nature du paramètre, nous avons relevé deux cas à considérer. Dans le premier cas, le poids ou l'importance du paramètre augmente en fonction de son ancienneté. C'est le cas du diabète, par exemple. Plus il est vieux, plus il a d'incidence sur les problèmes cardiaques du patient. Dans le deuxième cas, plus le paramètre est ancien, plus il perd de son importance. Un infarctus survenu il y a dix ans est certainement moins important qu'un récent infarctus. Donc, le temps peut ou bien diminuer le poids d'un paramètre ou au contraire l'augmenter.

La valeur d'un paramètre mesuré recèle deux informations distinctes. Elle vérifie d'un côté, la présence ou l'absence du paramètre mesuré, selon qu'il soit positif ou négatif. Et détermine, d'un autre côté, l'importance de la valeur du paramètre. Ces deux notions, bien qu'évidentes, doivent être capturées et bien modélisées. La valeur d'un paramètre est dans la plupart des cas moins importante que la présence ou l'absence du paramètre lui même. C'est pourquoi, la pondération doit pouvoir mettre en évidence la présence du paramètre. Dans le cas de notre algorithme, la distance entre deux valeurs considérées comme positives doit être clairement supérieure à la distance entre une valeur considérée positive et une autre considérée négative. Nous avons appliqué une formule simple qui satisfait toutes les remarques citées ci-dessus.

Présence :

Poids qui augmente avec le temps :

$$a^* \, ValeurAttributNormalisée + b^* NbrJoursAttribut / NbrMaxJours + 0,5 \quad (1)$$

Avec $a > 0$, $b > 0$ et $a + b = 0.5$

Poids qui diminue avec le temps :

$$a^* ValeurAttributNormalisée + b^* (NbrMaxJours - NbrJoursAttribut) / NbrMaxJours + 0,5 \quad (2)$$

ou $a*ValeurAttributNormalisée+b*(1- NbrJoursAttribut/NbrMaxJours) + 0,5$ (3)

Avec a > 0, b > 0 et a+ b = 0.5

Absence : *0,0*

ValeurAttributNormalisée est la valeur mesurée normalisée. *NbrJoursAttribut* est le nombre de jours depuis que le paramètre a été mesuré. *NbrMaxJours* est le nombre de jours correspondant à la plus ancienne date associée à l'attribut qu'on peut trouver dans la base de données.

Des formules sus-citées et des conditions qui leurs sont associées, on peut aisément démontrer que si la valeur d'un attribut existe pour un patient donné, elle est toujours supérieure à 0,5 et inférieure ou égale à 1. Alors que son absence génère une valeur nulle. Les paramètres a et b déterminent respectivement le poids qu'on veut donner à la valeur de l'attribut et au délai écoulé depuis sa mesure.

6.3.3. Détermination du nombre de clusters et création des groupes

L'inconvénient de la méthode des K-means est l'obligation pour l'utilisateur de déterminer, au préalable, le nombre de clusters sur lequel il veut diviser la population à modéliser. Le nombre n'est pas proposé par la méthode, mais doit être fourni comme paramètre par l'utilisateur. Les résultats dépendent grandement du choix fait. Deux individus peuvent se retrouver dans un même cluster si le nombre de clusters est $\mathcal{N}$, pour être dans deux clusters différents si le nombre est égal à $\mathcal{N}+1$, puis se retrouver une deuxième fois dans un même cluster si le nombre de clusters est de $\mathcal{N}+2$.

Pour remédier à ce problème nous avons proposé une méthode simple qui va aller chercher le nombre optimal de clusters à générer. Notre approche est basée sur la détermination du nombre de clusters en prenant un nombre de référence $\mathcal{R}$ qui représente dans les meilleurs des cas le nombre de clusters soupçonné ou supposé. La méthode des K-means va être appliquée sur la population étudiée n fois. Le nombre n est déterminé par l'utilisateur en prenant en compte la taille de la population, les ressources disponibles et le temps d'exécution souhaité. Plus grand est le nombre n plus précise est la méthode. Une première astuce de détermination du nombre n est de vérifier s'il y a un minimum ou maximum de clusters qu'il ne faut dépasser. S'il existe un maximum $\mathcal{M}$ à ne pas dépasser, le nombre n peut être estimé à $2*(\mathcal{M}-\mathcal{R})+1$. De la même façon, s'il existe un minimum m à ne pas dépasser, le nombre n peut être estimé à $2*(\mathcal{R}-m)+1$. S'il existe, à la fois, un minimum m et un maximum $\mathcal{M}$ à ne pas dépasser, le nombre n peut être estimé à $(\mathcal{M}-m)+1$. Nous pouvons d'ores et déjà faire la remarque qu'il existe au moins un minimum à ne pas dépasser qui est la valeur deux. Ces trois estimations du nombre n permettent d'appliquer la méthode des K-means un nombre de fois égal à gauche et à droite du nombre de référence $\mathcal{R}$.

Une fois les paramètres m et $\mathcal{M}$ estimés, Un premier ensemble de clusters est construit en posant le nombre de clusters égal à m. Puis un deuxième ensemble de $m+1$ clusters,...un $x^{ème}$ ensemble de $m+(x-1)$ clusters,...et finalement, un dernier ensemble de $\mathcal{M}$ clusters est construit. Les ensembles de clusters ainsi

constitués, sont utilisés pour grouper de nouveau les éléments de la population étudiée. L'idée est de regrouper ensemble les éléments de la population qui se sont retrouvés le plus souvent ensemble dans les clusters précédemment constitués. Pour cela un seuil α est défini. Il représente le nombre minimal de cooccurrences de deux éléments pour les affecter à un même cluster. Il n'existe pas de recommandations strictes ou une formule de détermination du seuil. Mais, nos expérimentations ont démontré que les meilleures performances sont obtenues avec un $\alpha = \mathcal{R}\text{-}1$. Un tel résultat n'est pas surprenant si nous examinons plus en détail la constitution des clusters définitifs.

Une première remarque est que le choix d'un α trop grand augmenterait le nombre de clusters de façon vertigineuse. En effet un α trop grand ne regrouperait dans un même cluster que les éléments quasi-identiques et génèrerait pour presque chaque élément un nouveau cluster.

Une deuxième remarque est que pour des valeurs de α légèrement inférieures ou égales à $\mathcal{R}$ le résultat final est sensiblement le même. Si deux éléments qu'on ne considère pas comme assez proche, se retrouvent dans un premier passage de l'algorithme, dans un même cluster à cause d'une valeur trop petite d'α, un passage ultérieur de l'algorithme rectifierai l'erreur en associant un des deux éléments à un troisième élément avec lequel le nombre de cooccurrences est plus grand qu'entre les deux premiers éléments. Le choix d'un α légèrement inférieure à $\mathcal{R}$ n'altère pas le résultat final, mais peut nuire considérablement aux performances de la méthode en augmentant le nombre de réaffectations.

Une troisième et dernière remarque est que le choix d'un α trop petit peut générer un trop petit nombre de clusters et ainsi être trop loin du nombre de référence $\mathcal{R}$.

En définitive, le seuil α ne doit être ni trop grand, pour ne pas générer un trop grand nombre de clusters, ni trop petit pour, d'une part, ne pas générer un nombre trop petit de clusters et d'autre part, ne pas augmenter le nombre de réaffectations et ainsi altérer les performances de la méthode. La valeur d'expérimentation de départ $\alpha = \mathcal{R}\text{-}1$ est tout indiquée. Il appartient à l'utilisateur de l'ajuster en fonction des résultats obtenus et ceux souhaités.

Une fois la méthode des K-means appliquée n fois et la valeur de α choisie, L'algorithme va affecter les éléments à traiter comme suit :

Créer le groupe ***GR(1)*** *et lui affecter le premier élément* Elt_1

Elt_1 ***[Groupe, Valeur, EltCooccurrent] = [(GR(1), 1, Elt_1)]***

Pour x = 2 à $\mathcal{N}$ *(*$\mathcal{N}$ *étant la taille de la population)*

Calculer cooccurrence (Elt_i, Elt_x*), i = 1 à x-1*

Si $\exists$ *i tel cooccurrence (*Elt_i, Elt_x*) >=* α *alors*

Choisir Elt_i *tel que cooccurrence (*Elt_i, Elt_x*) = Max (cooccurrence (*Elt_i, Elt_x*)) i = 1 à x-1*

Affecter Elt_x *au groupe* ***GR(i)*** *contenant l'élément* Elt_i

Elt_x***[Groupe, Valeur, EltCooccurrent] = [(GR(i),*** *cooccurrence(*Elt_i,Elt_x***), Elt_i)]***

Si Elt_i***[Valeur]<*** *cooccurrence(*Elt_i,Elt_x*) alors*

Elt_i***[Groupe, Valeur, EltCooccurrent] = [(GR(i),***cooccurrence(*Elt_i,Elt_x***), Elt_x)]***

Fin Si

Sinon

Créer un groupe ***GR(x)*** *et affecter* Elt_x *au groupe* ***GR(x)***

Elt_x ***[Groupe, Valeur, EltCooccurrent] = [(GR(x), 1, Elt_x)]***

Fin Si

Fin pour

6.4. Optimisation des performances

La méthode proposée ci-dessus a donné de très bons résultats. Elle regroupe ensemble les éléments qui se retrouvent le plus souvent dans un même groupe. Elle élimine par cette approche les imperfections qui peuvent s'introduire si nous utilisons les K-means avec un nombre fixe de clusters. Nous signifions par imperfections, les erreurs qui sont le résultat direct du choix du nombre de clusters. Par exemple, si nous appliquons la méthode des K-means à notre population de patients en considérant le nombre de clusters égal à deux, le résultat final est très influencé par le sexe des patients. Environ 70% des hommes sont affectés à un même groupe et 70% des femmes sont affectée au deuxième groupe, alors que le sexe n'est pas une caractéristique déterminante de détection et de traitement des maladies cardiovasculaires. Dans ce cas, notre choix de la valeur deux (2) a fortement favorisé un attribut parmi tous les autres. Notre méthode annihile ce genre d'erreurs en faisant varier le nombre de clusters.

Même si la méthode proposée utilise les K-means n fois, ses performances, à ce niveau, restent toujours supérieures aux méthodes statistiques qui ont souvent un temps d'exécution exponentiel en fonction de la taille de la population étudiée. Les performances de notre méthode commencent à se détériorer lors des traitements supplémentaires d'affectation des éléments à leurs nouveaux groupes constitués à partir des premiers résultats. Les tests de comparaison lors du calcul des cooccurrences forment une suite arithmétique dont la somme est égale à $n * \mathcal{N} * (\mathcal{N}+1) / 2$ où n est le nombre de fois où les K-means sont utilisés et $\mathcal{N}$ est la taille de la population étudiée. Les tests entre les valeurs des cooccurrences pour le choix de la plus grande valeur forment eux aussi une suite arithmétique dont l'expression est $(\mathcal{N}-2) * (\mathcal{N}-1)/2$. À une fin d'illustration, pour traiter une population de $\mathcal{N}$ = 10.000 éléments et n = 10, le nombre de tests de

calcul des cooccurrences équivaudrait à **500 050 000** et le nombre de tests entre les valeurs des cooccurrences serait égal à **49 985 001**. Soit plus d'un demi milliard de tests pour une population moyenne de dix mille éléments. En terme de temps d'exécution, le traitement des 2400 patients de notre étude prenait plusieurs heures.

Afin de remédier à cette contre-performance, nous devions réduire le temps d'exécution en diminuant intelligemment le nombre de tests. Notre solution était de ne pas effectuer tous les tests, en s'arrêtant au premier élément qui satisfait à la condition de la cooccurrence. Pour affecter un élément Elt_x à un groupe, nous ne le comparons plus à tous les éléments Elt_i i=1 à x-1, mais nous nous arrêtons au premier élément qui satisfait la condition «cooccurrence (Elt_x, Elt_i) >=α ». L'algorithme précédent est transformé comme suit :

Créer le groupe **GR(1)** *et lui affecter le premier élément* Elt_1
Pour $x = 2$ *à* $\mathcal{N}$
    Pour $i = 1$ *à* $x-1$
        Calculer cooccurrence (Elt_i, Elt_x)
        Si cooccurrence (Elt_i, Elt_x) >= α *alors*
            Affecter Elt_x *au groupe* **GR(i)** *contenant l'élément* Elt_i
            Sortir du pour i
        Fin Si
    Fin pour i
    Si Elt_x *n'est pas encore affecté à un groupe alors*
        Créer un groupe **GR(x)** *et affecter* Elt_x *au groupe* **GR(x)**
    Fin Si
Fin pour x

L'avantage d'un tel algorithme est qu'il réduit considérablement le nombre de tests. Si les éléments de départ sont bien choisis, l'algorithme peut donner un résultat final presque équivalent au premier algorithme avec un temps de traitement beaucoup plus petit. Son inconvénient est qu'il peut donner de très mauvais résultats si α est trop petit et les éléments de départ sont mal choisis.

Chacun des éléments de départ doit être le plus représentatif possible d'un des clusters finaux, pour que les bons clusters soient formés le plus tôt possible, augmentant ainsi les chances des éléments qui viennent ensuite d'être affectés au bon cluster. Comme il est difficile de savoir si un élément est représentatif d'un cluster final qui n'est pas encore formé, nous recommandons de commencer avec les centroides du cluster construit avec un nombre de clusters de référence, soit $n = \mathcal{R}$. Dans un cas optimal où les éléments de départ sont les futures centroides des clusters qui seront construits, le nombre de tests de comparaison lors du calcul des cooccurrences ne serait plus influencé par le carré de la taille de

la population et pourrait être majoré à $\mathcal{R}'*(\mathcal{R}'+1) / 2 + (\mathcal{N}-\mathcal{R}')*\mathcal{R}'$. Où $\mathcal{R}'$ est le nombre de groupes construits. Les tests entre les valeurs des cooccurrences ne sont plus indispensables.

Dans nos expérimentations, le deuxième algorithme a donné exactement le même résultat que le premier pour $n=5$ et sans qu'on ait effectué un choix sélectif des éléments de départ.

6.5. Affectation flou en définissant un deuxième seuil β

Il est aussi possible, à partir du premier algorithme, d'affecter un élément à plus d'un cluster. Il suffit pour cela de construire les clusters de la même façon que dans la méthode standard. Puis, un deuxième seuil $\beta<\alpha$ est défini. Pour chaque élément Elt_i de la population à traiter et dans chaque cluster final auquel n'appartient pas Elt_i, on vérifie s'il existe un élément Elt_j tel que cooccurrence $(Elt_i\ Elt_j)>=\beta$. Si c'est le cas, l'élément Elt_i est affecté au groupe auquel appartient l'élément Elt_j avec une certaine valeur de pondération. L'algorithme de départ est transformé comme suit :

Créer le groupe **GR(1)**
Affecter le premier élément Elt_1 *au groupe* **GR(1)** *avec une pondération égale à 1*
Elt_1 **[Groupe, Valeur, EltCooccurrent, Pondération] = [(GR(1), 1, Elt_1),1]**
Pour x = 2 à $\mathcal{N}$
  Calculer cooccurrence (Elt_i, Elt_x), *i = 1 à x-1*
  Si $\exists$ *i tel cooccurrence* $(Elt_i, Elt_x) >= \alpha$ *alors*
    Choisir Elt_i *tel que cooccurrence* (Elt_i, Elt_x) = *Max (cooccurrence* (Elt_i, Elt_x)*) i = 1 à x-1*
    Affecter Elt_x *au groupe* **GR(i)** *contenant l'élément* Elt_i
    Elt_x**[Groupe, Valeur, EltCooccurrent, Pondération]** = **[(GR(i),** *cooccurrence*(Elt_i,Elt_x)**, Elt_i),1]**
    Si Elt_i**[Valeur]<** *cooccurrence*(Elt_i,Elt_x) *alors*
      Elt_i**[Groupe, Valeur, EltCooccurrent, Pondération]** = **[(GR(i),**$cooccurrence(Elt_i,Elt_x)$**, Elt_x),1]**
    Fin Si
  Sinon
    Créer un groupe **GR(x)** *et affecter* Elt_x *au groupe* **GR(x)**
    Elt_x **[Groupe, Valeur, EltCooccurrent, Pondération] = [(GR(x), 1, Elt_x),1]**
  Fin Si
Fin pour x

Pour i=1 à $\mathcal{N}$ *faire*
  De chaque groupe **GR(j)** $\neq$ **GR(i)** *choisir un élément* Elt_j *qui satisfait les deux conditions suivantes*
    cooccurrence $(Elt_i, Elt_j)>=\beta$ *et*
    cooccurrence (Elt_i, Elt_j) *est la cooccurrence maximale que peut avoir* Elt_i *dans le groupe*
  Affecter Elt_i *à* **GR(j)**
  Fin chaque
  Pour chaque groupe **GR(j)** *contenant* Elt_i

Calculer la pondération $\mathcal{P}$ = *cooccurrence* (Elt_i, Elt_j) / *somme(cooccurrence* (Elt_i))

Fin chaque

Fin Pour i

7. RÉSULTATS ET INTERPRÉTATION

Pour l'implémentation de notre méthode, nous avons choisi la population de patients comme expliqué plus haut, sur laquelle nous avons appliqué les deux algorithmes, avant et après optimisation. Les résultats des deux approches sont quasi-identiques. L'algorithme simplifié de la méthode optimisée est le suivant :

Pour i = 2 à 8

Utiliser les K-means pour grouper les patients

Fin Pour

Créer le groupe ***GR1*** *et lui affecter le premier patient*

De 2 jusqu'au dernier patient faire

Si *Nouveau Patient a été dans le même groupe qu'un ancien patient plus de 04 fois alors mettre Nouveau patients dans le même groupe que l'ancien*

Sinon *Créer un nouveau groupe*

Affecter Nouveau Patient à un Nouveau Groupe

Fin Si

Fin Jusqu'à

Les paramètres choisis sont :

- *Le nombre de clusters de référence* $\mathcal{R}$: Notre population choisie est constituée de patients cardiaques ayant une maladie coronarienne. Nous avons donc, soupçonné l'existence de groupes homogènes de malades souffrant de maladies similaires et ayant subit des traitements équivalents. Nous avons recensé quatre types de maladies coronariennes que sont l'angine instable (AI), l'infarctus (IM), les maladies coronariennes athérosclérotiques (MCAS) et les cardiopathies ischémiques. Auxquelles, maladies nous avons ajouté le terme autre (AUT) pour désigner les autres maladies de type cardiaque mais qui ne sont pas coronariennes. Nous avons, alors choisi un nombre de clusters de référence égal à cinq. Soit $\mathcal{R}$ = 5.
- *Le nombre minimal de clusters de départ* m: Comme on avait aucune indication sur le seuil inférieur du nombre de clusters, nous avons pris le plus petit minimum admissible, soit deux. m = 2.
- *Le nombre maximal de clusters à atteindre* $\mathcal{M}$: Comme nous avons pu déterminer $\mathcal{R}$ et m, nous pouvions alors calculer $\mathcal{M} = \mathcal{R} + (\mathcal{R}-m) = 2*\mathcal{R}-m$ = 2*5-2 = 8. Soit $\mathcal{M}$ = 8.
- *Le nombre de fois où les K-means sont utilisés* n : $n = (\mathcal{M}-m)+1$ = 8-2+1 = 7. Soit n = 7.
- *Le seuil de cooccurrence* α : La valeur recommandée est $\alpha = \mathcal{R}-1$ = 5-1 = 4. Soit α = 4.

La méthode a donné les résultats résumés dans le *tableau 16* et la *figure 18*.

Groupe	01		03	04	06
Nombre Patients	471		18	421	22
Sous Groupe	451	20	18	421	22
ECG	1	0	0	1	0
Troponine	0,95	-1	-1 ou 1	1	-1
CK-MB	généralement pas fait	Pas fait	pas fait ou non pertinent	60	20
Hypercholestérolémie	74,72%	50,00%	94,44%	63,66%	45,45%
Hypertension Artérielle	60,98%	35,00%	77,78%	42,36%	40,91%
Maladies Coronariennes	43,24%	0,00%	100,00%	18,55%	22,73%
Diabète	26,61%	15,00%	44,44%	25,31%	22,73%
Autres Facteurs de Risque	7,32%	0,00%	0,00%	4,76%	0,00%

Tableau 16: Groupes crées par la méta-méthode multi K-means

Caractéristiques des groupes de patients

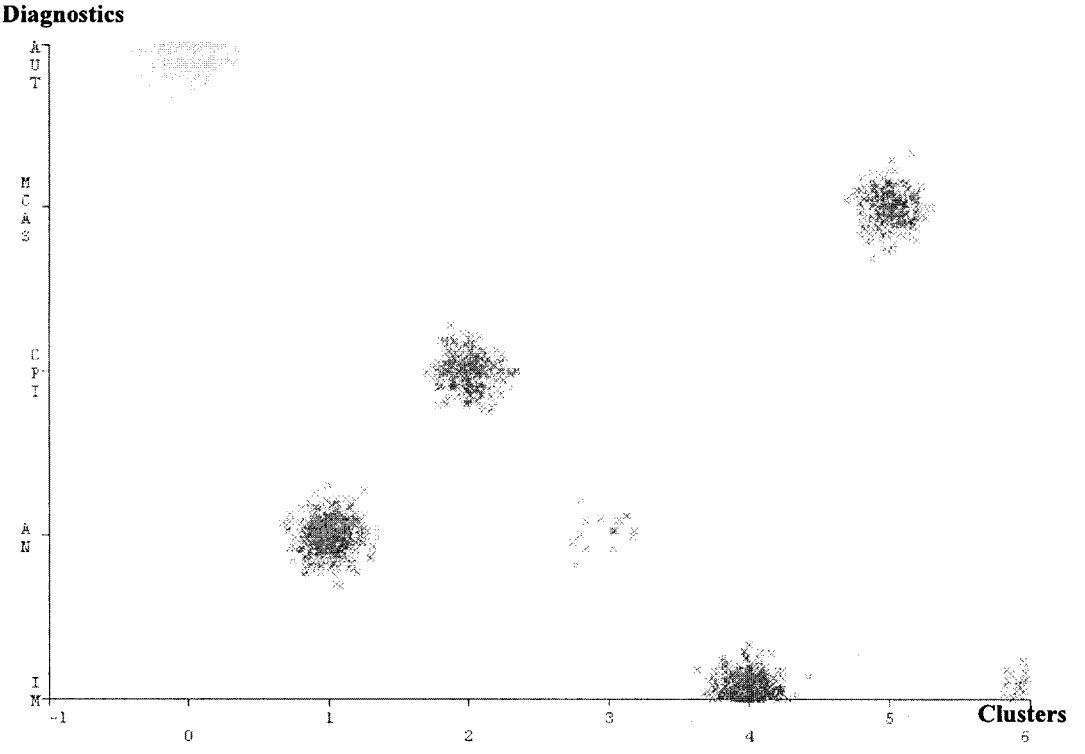

Figure 18 : Répartition de la population de patients

Le *tableau 16* et la *figure 18*, sont présentés pour commenter les résultats et faciliter leur interprétation. Dans le *tableau 16* ne sont reportés que les groupes ayant besoin d'explication. Nous commençons par la *figure 18* qui montre clairement que les patients sont répartis en fonction des maladies coronariennes dont ils sont atteints. Seuls les diagnostics principaux et les clusters sont représentés dans la figure. Comme les éléments d'un même cluster ont un même diagnostic principal, ils ne devraient être représentés que par un unique point. Nous avons volontairement dispersé les patients appartenant à un même cluster, sous la forme d'un nuage, pour que la figure soit plus explicite. La création du nuage autour du point représentant le cluster se fait en fonction des autres attributs. Pour deux patients différents, plus les valeurs des attributs les constituant sont proches plus ils sont proches dans le nuage de point.

Plus Les groupes numérotés 0, 2 et 5 ne souffrent d'aucun problème d'interprétation. Ils représentent respectivement les patients souffrant de maladies cardiaques non coronariennes, les patients ayant une cardiopathie ischémique et les patients souffrant de maladie coronarienne athérosclérotique. Les groupes numérotés 1, 3, 4 et 6 ont besoin de plus d'explications. La question est pourquoi des patients souffrant d'une même maladie se retrouvent séparés dans deux groupes différents. Plus précisément, qu'est ce qui caractérise les quelques patients du groupe 3 et 6 pour qu'ils soient mis seuls dans un groupe. Les caractéristiques que nous avons relevées dans le *tableau 16* devraient nous éclairer.

Le groupe 3 a une forte particularité qui le caractérise. Ses 18 patients n'ont pas subi d'ECG, lequel reste un test systématique dans de tels cas. C'est une première caractéristique qui a fait que ces malades souffrant d'angine ont été séparés des autres. Mais cela ne suffit pas, parce qu'il existe 20 autres cas de patients souffrant d'angine, qui n'ont pas subi d'ECG et qui sont quand même dans le même groupe que les autres. Ce qui différencie nos 18 patients des 20 autres est qu'ils ont de lourdes prédispositions aux maladies cardiovasculaires. 94,44% d'entre eux ont un problème d'hypercholestérolémie et 77.78% d'hypertension artérielle contre seulement 50,00% d'hypercholestérolémie et 35,00% d'hypertension artérielle pour les autres 20 patients. Mais surtout, les 18 patients, représentant le groupe 03, ont tous à 100% été diagnostiqués comme coronariens dans le passé alors que les 20 patients, du groupe 01, n'ont jamais eu de problèmes coronariens connus. L'algorithme les a alors considérés comme faisant parti du groupe des patients souffrant d'angine. Dans la pratique, nous comprenons que les 18 patients qui se sont présentés pour un problème coronarien et qui avaient des antécédents de diagnostics coronariens et qui n'ont pas subi d'ECG font ressortir de graves erreurs que nous soupçonnons dans la saisie de l'information plutôt que dans la pratique médicale.

Le groupe 6, de même que le 4, contiennent les patients ayant eu un infarctus. Les 22 patients du groupe 6 ont été séparés des 399 patients du groupe 4. La seule différence entre les deux groupes est que les 22 patients n'ont pas subi d'ECG à leurs admissions à l'hôpital ce qui n'est pas normal pour des patients ayant eu un infarctus. Une analyse[10] plus poussée a montré que ces 22 patients ont été transférés d'autres hôpitaux. Ils avaient donc subi un examen d'ECG avant d'être admis à l'hôpital de Fleurimont. La

[10] Analyse et interprétation faites par Docteur Michel Nguyen Professeur de Cardiologue au CHUS de Fleurimont

méthode, bien qu'elle ne connaisse pas l'importance de l'ECG dans le diagnostic des maladies cardiaques, a ressorti une anomalie très importante. Heureusement pour la médecine et la réputation de l'établissement hospitalier, l'erreur est d'ordre logistique.

Une dernière remarque, est que la méthode n'a, à aucun moment, mis dans un même groupe deux patients ayant deux diagnostics principaux différents. Elle n'a fait que séparer les patients ayant un même diagnostic. Elle est, donc, allée faire ressortir les particularités de certains éléments, qui semblaient, a priori faire partie d'un groupe homogène.

8. COMPARAISON ET ÉVALUATION

Pour mettre en évidence l'apport de la méta-méthode, nous ne pouvions trouver mieux que de comparer ses résultats à ceux des K-means de base. L'apport de la méta-méthode peut être mesuré par la qualité des résultats comparativement à la méthode des K-means de base et aussi par les performances d'exécution une fois les optimisations réalisées.

La non présence d'ECG dans le cas des 22 patients ayant eu un infarctus a surpris le médecin spécialiste qui faisait l'interprétation. L'analyse, qui en a été faite, a soulagé tout le monde. La méthode a permis de mettre le point sur une erreur d'ordre logistique qui pouvait fausser les études sur la pratique médicale établie au CHUS. La méta-méthode a, en plus de bien grouper les éléments de la population de patients, fait ressortir des caractéristiques insoupçonnées jusqu'alors. Elle reste un bel exemple sur l'importance des méthodes d'exploration, lesquels, et à la différence des méthodes de prédiction et d'estimation qui ne font que reproduire des caractéristiques déjà connues de la population à étudier, peuvent produire des résultats totalement imprévisibles qu'ils faut interpréter.

Pour illustrer les améliorations apportées par la méthode, nous reproduisons ci-dessous les résultats des K-means de base avec un nombre de clusters égal à 5. Nous y confrontons les groupes formés avec les diagnostics des patients.

Cluster	MCAS	Cardiopathie Ischémique	Angine Instable	Infarctus	Autres	Total
00	7	47	38	21	150	**263**
01	86	102	160	146	273	**767**
02	84	132	159	242	255	**872**
03	0	0	0	0	323	**323**
04	69	31	132	35	219	**486**
TOTAL	**246**	**312**	**489**	**444**	**974**	

Tableau 17: Groupes crées par la méthode des K-means de base

Groupes en fonction des diagnostics

- 533 des 974 patients dont les diagnostics principaux ne sont pas spécifiques ont un diagnostic secondaire coronarien
- 246 patients ont un deuxième diagnostic coronarien
- Tous les patients du groupe 03 ne sont pas hypertendus. Soit 872 patients sur les 1189 non hypertendus.
- Tous ceux du groupe 02 sont hypertendus. Soit 767 patients sur les 1276 hypertendus.

Il devient clair, en se basant sur les remarques ci-dessus, que la méthode des K-means n'a pas groupé les patients selon leurs diagnostics principaux, mais bien sur d'autres caractéristiques que nous résumons dans le *tableau 18*.

Cluster	Nbr	Pas ECG		Avec HTA		Sans HTA		Non coronarien		Antécédent coronarien	
00	256	**256**	**79,01%**	142	11,13%	114	9,59%	3	0,92%	129	13,48%
01	681	0	0,00%	**681**	**53,37%**	0	0,00%	0	0,00%	196	20,48%
02	788	0	0,00%	0	0,00%	**788**	**66,27%**	0	0,00%	176	18,39%
03	323	60	18,52%	133	10,42%	190	15,98%	**323**	**98,78%**	39	4,08%
04	417	8	2,47%	320	25,08%	97	8,16%	1	0,31%	**417**	**43,57%**
Totaux	**2465**	**324**		**1276**		**1189**		**327**		**957**	

Tableau 18: Groupes crées par la méthode des K-means de base

Groupes en fonction des caractéristiques

Le groupe 00 contient, majoritairement, les patients n'ayant pas effectués le test d'ECG. Alors que plus de la moitié de la composante du groupe 01 est constituée de patients ne souffrant pas d'hypertension artérielle. Les deux tiers du groupe 02 ont un problème d'hypertension et la presque totalité du groupe 03 ont un diagnostic principal non coronarien. Alors que le groupe 04 est constitué à moitié de patients ayant un antécédent coronarien. Ni la troponine, ni la CK-MB, qui sont des tests spécifiques pour la détection des maladies coronariennes, n'ont été ressorti par la méthode. Cette dernière a bien constitué un groupe de patients de non coronariens, identifié comme *«autre»*. Les autres types de diagnostics ne sont pas ressortis en tant que groupes. La raison principale est que les maladies coronariennes sont très proches du point de vue de leurs caractéristiques. Il arrive, dans la pratique, qu'un médecin ne puisse pas aboutir à un diagnostic définitif dans les cas d'angine instable ou d'infarctus. Aussi, Le MCAS est un terme général qui peut regrouper des maladies coronariennes non identifiées ou non encore connues. Il convient aussi de rappeler, comme mentionné dans les deux remarques plus haut, que la duplicité de

diagnostics de type coronarien et la simultanéité de deux types de diagnostics, l'un coronarien et l'autre pas, ajoute aussi de l'ambiguïté au statut final du patient. La méthode a, alors, favorisée d'autres caractéristiques du patient cardiaque que sont le test d'ECG, les facteurs de risque (hypertension) et les antécédents coronariens.

Notre méta-méthode a rectifié les lacunes rencontrées, en allant chercher des caractéristiques autres que celles favorisées par le nombre de clusters égal à cinq. Sachant qu'en faisant varier le nombre de clusters, la méthode des k-means favorise à chaque fois des caractéristiques différentes. En synthétisant les résultats, elle aboutit à un résultat quasi parfait.

9. CONCLUSION

Afin de mener à bien notre travail qui consistait en la classification non supervisée de patients cardiaques, nous sommes passés par différentes étapes préliminaires. Nous avons commencé par définir puis choisir la population à étudier. Nous avons, ensuite, effectué des traitements de nettoyage et de transformation des données caractérisant cette population. Après la préparation des données, nous avons développé une méta-méthode que nous avons appelé Multi K-means, laquelle repose sur la combinaison des résultats de l'utilisation d'un nombre limité de fois des K-means avec différentes valeurs de clusters en sortie. À la fin, nous avons comparé les résultats de notre méta-méthode avec ceux des K-means de base.

Cette comparaison nous a permis de conclure que la méta-méthode a donné des résultats très intéressants et beaucoup plus faciles à interpréter que ceux des K-means. Elle a rehaussé la qualité des résultats, en favorisant des caractéristiques autres que celles ressorties en prenant un nombre de clusters fixe et égal à cinq, dans notre cas. En faisant varier le nombre de clusters, la méthode des K-means favorise à chaque fois des caractéristiques différentes. La méta-méthode synthétise ces résultats en les confrontant les uns aux autres afin d'aboutir à une solution quasi parfaite. Un autre apport très important de la méta-méthode est que celle-ci détermine d'elle-même, selon les paramètres en entrée, le nombre final de clusters. Les connaissances préliminaires et souvent intuitives de l'utilisateur ne constituent plus une donnée en sortie mais bien un simple paramètre de départ.

Il convient de remarquer que la méthode des K-means de base peut, elle aussi, révéler des modèles de données intéressants. Il incombe à l'utilisateur et aux spécialistes du domaine d'en faire l'interprétation, même si celle-ci n'est pas évidente. L'intérêt des méthodes de groupage de données est bien dans la révélation de nouveaux modèles plutôt que dans la vérification de modèles qu'on soupçonne dès le départ.

Conclusion et perspectives

Chapitre V

1. CONCLUSION ET PERSPECTIVES

Nous avons proposé une nouvelle méta-méthode pour la classification non supervisée des données provenant d'une base de données médicale. Cette approche utilise les sorties de la méthode des k-means appliquée plusieurs fois aux données avec différents nombres initiaux de clusters. Elle combine les résultats pour d'une part diminuer le risque d'erreur qui est inhérent à l'utilisation d'un nombre unique et précis de clusters et d'autre part, augmenter la qualité de la classification en ne classant deux éléments dans un même groupe que si cela est confirmé un nombre de fois prédéterminé. Nous résumons ci-dessous, les différentes étapes par lesquelles nous sommes passés et la contribution apportée par notre approche.

1.1. Conclusion

Notre objectif principal restait la détermination de sous-groupes dans une population initiale de patients coronariens en utilisant la classification non supervisée. La détermination d'une telle population exigeait une définition claire du patient cardiaque. À des fins de simplification et de commodité, nous avons convenu que notre population était formée de tous les patients admis au service de cardiologie pendant l'année 2002. Pour faciliter la détermination des données à extraire, nous les avons auparavant, regroupées en six classes selon leurs types. Les données personnelles, les tests, les diagnostics, les traitements, les antécédents de diagnostics et les antécédents de traitements. Une fois, les classes de données déterminées nous avons pu identifier et extraire les variables les plus pertinentes en fonction de nos objectifs.

Les données brutes ainsi extraites avaient besoin de nettoyage et de transformation pour être utilisables. Nous avons identifié trois grandes classes de données à nettoyer. Les diagnostics et les traitements, les données personnelles et les tests. Pour la première classe citée, nous n'avons effectué aucun traitement de correction parce que ce type de données reste très fiable et nous ne voulions pas altérer la qualité des données en entrée en apportant des modifications considérées comme majeures et qui toucheraient le diagnostic ou le traitement. Pour les données personnelles, les seules erreurs provenaient des redondances dues à la forme féminine, la forme plurielle, la forme fléchie, les synonymes, et les fautes d'orthographe. Nous avons réglé le problème par l'adoption d'un dictionnaire recensant tous les cas rencontrés et pouvant être enrichi par les cas futurs. Le nettoyage des données de tests restait la partie la plus délicate de cette étape. Une erreur pouvait survenir à la suite d'une maladresse humaine ou à l'utilisation de différents appareils et procédés de test et de mesure pour déterminer l'évaluation d'un seul phénomène. Nous avons proposé six types de prétraitement sur les tests. L'identification des formats, le traitement des erreurs, la suppression des unités de mesure, le traitement des dates, le traitement des remarques et le traitement des valeurs manquantes. Pour chaque type de traitement nous avons proposé une solution immédiate et éventuellement une proposition de solution future pour une prise en charge plus efficace de telles erreurs.

Après le nettoyage, nous avons procédé à des transformations de l'état initial des données pour les adapter aux algorithmes de datamining et espérer de meilleurs résultats. Nous avons, alors, effectué deux types de transformations sur les données, le regroupement des diagnostics et des traitements par type et la discrétisation de certains tests tels que la Troponine et l'ECG. Le regroupement avait pour but d'absorber les détails et préparer le terrain aux méthodes de datamining pour un plus grand degré de généralisation. Alors que la discrétisation permettait la récupération des seuils de tests établis par la pratique. Elle simplifiait le traitement des données par les méthodes de datamining et aussi facilitait grandement l'interprétation des résultats en récupérant les repères sur lesquels se basent les experts du domaine.

Pour l'exploration des données, nous avons opté pour la classification non supervisée pour essayer de classer la population de départ en groupes ayant une signification particulière et pouvant mieux nous renseigner sur les patients étudiés. Notre choix des k-means s'est fait pour la simplicité de la méthode, la qualité des résultats qu'elle produit et surtout pour ses temps d'exécution record. Avant d'appliquer la méthode, nous avons choisi avec soin les attributs les plus pertinents pour notre étude. Ces attributs devaient résumer le mieux possible l'hospitalisation d'un patient. Ils comprenaient les informations personnelles, la prise en charge, les tests subis, les diagnostics, les diagnostics antécédents, les traitements et les traitements antécédents. Nous devions aussi résoudre le problème de l'influence du temps sur le poids de certains antécédents. C'est le phénomène de l'usure par le temps. Comme il peut exister le phénomène inverse qui fait que le poids d'un antécédent est proportionnel à son temps de vieillissement. Nous avons résolu le problème en proposant une fonction simple qui va capturer la notion de temps et pondérer l'attribut concerné. Cette fonction de capture de la notion de temps nous a permis de différencier entre deux même valeurs d'un même attribut qui ont été recueillies à deux moments différents. Une fois tous ces préparatifs accomplis, nous avons appliqué notre méta-méthode sur les données d'entrée pour les assembler en un nombre initialement inconnu de groupes que la méthode détermine d'elle même. L'idée principale de la méta-méthode est d'appliquer la méthode des k-means un nombre n de fois en faisant varier le nombre de clusters, puis classer ensemble les individus qui se sont retrouver le plus grand nombre de fois dans un même cluster. Cette idée toute simple, permet de minimiser les erreurs inévitables dues à une seule utilisation de la méthode des k-means avec un nombre unique de clusters au départ. Elle permet aussi, d'augmenter considérablement l'efficacité de la méthode en ne mettant ensemble deux individus que si ceci est confirmé un nombre prédéterminé de fois.

1.2. Perspectives

Notre travail a été fait dans un cadre pratique qui nous a permis de soulever la difficulté et la complexité de traiter des données réelles qu'elles soient structurées ou pas. Ça nous a aussi donné l'opportunité de relever certains aspects théoriques relatifs aux fonctions de distance et au paramétrage de la méthode des k-means. Nous reproduisons ci-après le complément de travail qui devrait être fait pour une exploitation optimale des données et les quelques perspectives de recherche qui nous semblent important d'aller explorer :

- Traitement des erreurs : La meilleure façon de traiter les erreurs est de les éviter à la source. Il est important de distinguer les erreurs de saisie manuelle qu'on peut diminuer sensiblement en proposant des masques de saisie adéquats, en développant une assistance et aide à la saisie et en sensibilisant et formant le personnel. Le deuxième type d'erreurs est dû à la saisie automatique via les appareils électroniques. L'idéal serait qu'il y ait une harmonisation des unités de mesure et des formats de données sur tous les appareils et instruments de mesure. La deuxième solution envisagée est de traiter les données, par l'ordinateur connecté à l'appareil électronique, immédiatement après leurs captures. Cela à l'avantage de distribuer le temps de traitement sur plusieurs ordinateurs au fur et à mesure de l'acquisition des données. Son principal inconvénient est la nécessité d'installer les procédures de traitement des données sur tous les ordinateurs responsables de l'acquisition avec tout ce que cela implique comme déploiement initial et éventuellement, mises à jours ultérieures. La troisième solution est d'inclure le traitement des erreurs dans la phase de prétraitement que tout entreposage de données nécessite. Cette tâche alourdirait le processus, déjà complexe, de mise à jour de l'entrepôt de données. Mais serait d'une grande utilité pour les travaux futures. Un entrepôt de données n'a pas pour but final l'archivage des données mais leur analyse et exploration.

- Traitement des rapports : Un rapport est composé de faits, remarques, descriptions et d'éventuelles conclusions qu'un médecin rédige dans un langage naturel. Nous avons proposé une solution qui traite les rapports dans leur état actuel. Pour une marge d'erreur nulle dans le traitement des rapports, il est plus judicieux d'opter pour la standardisation de leurs contenu, soit, une intervention pré-rédaction. Mais, cela ne serait possible que par l'implication des médecins.

- Variantes de la méta-méthode: Une première variante serait de remplacer les K-means par d'autres méthodes de clustring. Il est aussi possible de combiner les résultats de plusieurs méthodes avec divers nombres de groupes. Les stratégies de combinaison restent à définir. Les résultats seront intéressants à analyser et leurs comparaisons feraient un bon sujet d'étude.

- Optimisation des performances : Le temps d'exécution est souvent critique dans les méthodes de datamining. Les méta-méthodes n'en sont que plus affectées. Notre proposition de réduire le nombre de tests a donné des résultats encourageants. Néanmoins, une étude plus stricte sur les conséquences d'une telle approche est nécessaire. Il est aussi possible de réduire le nombre de

tests en diminuant la taille de la population à étudier. L'une des solutions est de procéder de façon graduelle. On peut, par exemple, commencer par créer les clusters finaux immédiatement après que la méthode de base a été appliquée α fois. Si des éléments de la population ont une cooccurrence égale à α, ils sont affectés à un même cluster final. À la prochaine application de la méthode de base, ils sont remplacés par un seul représentant, mais ils auront un poids proportionnel à leur nombre. Leurs représentant peut être leur centroide ou un élu parmi eux.

- Affectation floue : Elle reste une bonne variante à étudier pour le cas où l'affectation d'un élément à un cluster n'est pas exacte et est pondérée par un degrés d'appartenance.

Références

[Abiteboul97] Serge Abiteboul , "Querying Semi-Structured Data", International Conference on Database Theory, Janvier 1997.

[Agarwal96] Sameet Agarwal, Rakesh Agrawal, Prasad M. Deshpande, Ashish Gupta, Jeffrey F. Naughton, Raghu Ramakrishnan, Sunita Sarawagi, "On the Computation of Multidimensional Aggregates", Proc. 22nd Int. Conf. Very Large Databases, VLDB, 1996.

[Agrawal98] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, Prabhakar Raghavan, "Automatic subspace clustering of high, dimensional data for data mining applications", Proceedings of ACM SIGMOD, International Conference on Management of Data, 1998.

[Alsabti99] Khaled Alsabti, Sanjay Ranka, Vineet Singh, "An Efficient K-Means Clustering Algorithm", PAKDD, page 355-359, 1999.

[Andrews95] R. Andrews, J. Diederich, A. Tickle, "A Survey and Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks", Knowledge-Based Systems, 1995.

[Atkinson89] Atkinson M., Bancilhon F., Dewitt D., Dittrich K., Maier D., Zdonick S. "The object oriented Data Base System Manifesto", Deductive and Object Oriented Databases International Conference, Kyoto, Japan,1989.

[Atkinson00] Malcolm P. Atkinson: Persistence and Java - A Balancing Act. Objects and Databases 2000: 1-31, Objects and Databases: International Symposium, Sophia Antipolis, France, June 2000.

[Barioni02] Maria Camila Nardini Barioni, Humberto Luiz Razente, Caetano Traina Jr., Agma J. M. Traina, "Visually Mining on Multiple Relational Tables at Once", ADBIS Research Communications, page 21-30, 2002.

[Benhadid98] Ilham Benhadid, Jean-Guy Meunier, Saâd Hamidi, Zira Remaki, Moses Nyongwa, "Étude Expérimentale Comparative des Méthodes Statistiques pour la Classification des Données Textuelles", JADT, 1998.

[Berkin02] Pavel Berkhin, "Survey of clustering data mining techniques", Technical report, Accrue Software, San Jose, California, 2002.

[Blum97] A. L. Blum, P. Langley, "Selection of Relevant Features and Examples in Machine Learning", Artificial Intelligence, vol. 97, page 245-271, 1997.

[Bottou95] Léon Bottou, Yoshua Bengio, "Convergence Properties of the K-Means Algorithms", Advances in Neural Information Processing Systems, 1995.

[Bradley98] P. S. Bradley, Usama M. Fayyad, "Refining Initial Points for K-Means Clustering", Proc. 15th International Conference on Machine Learning, 1998.

[Cabena98] Peter Cabena, Pablo Hadjinian, Rolf Stadler, Jaap Verhees, Alessandro Zanasi "Discovering data mining: from concept to implementation", Upper Saddle River (New Jersey): Prentice Hall, 1998.

[Caetano00] Caetano Traina Jr., Agma Traina, Leejay Wu and Christos Faloutsos, "Fast feature selection using the fractal dimension", XV Brazilian Symposium on Databases (SBBD), Paraiba, Brazil, October 2000.

[Chaudhuri97] Surajit Chaudhuri, Umeshwar Dayal, "An Overview of Data Warehousing and OLAP Technology ", SIGMOD Record 26 (1), page 65-74, 1997

[Chaudhuri98] Surajit Chaudhuri, "An overview of query optimization in relational systems", Symposium on Principles of Database Systems, Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, Seattle, Washington, United States, page 34-43, 1998.

[Chen97] Ming-Syan Chen, Jiawei Han, Philip S. Yu, "Data Mining: An Overview from Database Perspective", Ieee Trans., On Knowledge And Data Engineering, 1997.

[Davidson04] I. Davidson, Ashwin Satyanarayana, "Speeding up K-means clustering by Bootstrap averaging", Workshop on Clustering Large Data Sets, IEEE ICDM 2004.

[Dehne02] Frank Dehne, Todd Eavis, Susanne Hambrusch, Andrew Rau-Chapin, "Parallelizing the Data Cube", Lecture Notes in Computer Science, 2002.

[Dhillon00] Inderjit Dhillon, Dharmendra Modha, "A Data-Clustering Algorithm on Distributed Memory Multiprocessors", Lecture Notes In Computer Science, Volume 1759, Large-Scale Parallel Data Mining, Workshop on Large-Scale Parallel KDD Systems, SIGKDD, page 245-260, 2000.

[Dougherty95] James Dougherty, Ron Kohavi, Mehran Sahami, "Supervised and Unsupervised Discretization of Continuous Features ", International Conference on Machine Learning,1995.

[Fayyad96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, "from Data Mining to Knowledge Discovery: An Overview", Advances in knowledge discovery and data mining, Usama M. Fayyad & al. eds, page 1-34, 1996.

[Freitas96] AA. Freitas , SH. Lavington, "Speeding up knowledge discovery in large relational databases by means of a new discretization algorithm", *Advances in Databases (Proc 14th British Nat Conf on Databases, Edinburgh, UK)*, R. Morrison and J. Kennedy editors, pages 124-133, Berlin, 1996.

[Gardarinbd99] Georges Gardarin, " Base de Données objet & relationnel", Librairie Eyrolles, 1999.

[Gardarinii99] Georges Gardarin, " Internet / intranet et bases de données", Librairie Eyrolles, 1999.

[Goglin01] Jean-François Goglin, "Le datawarehouse pivot de la relation client ", Édition Hermes Science, 2001.

[Gray95] Jim Gray, Adam Bosworth, Andrew Layman, Hamid Pirahesh, J. Data Mining and Knowledge Discovery "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals", 1995.

[Harinarayan96] Venky Harinarayan , Anand Rajaraman, Jeffrey D. Ullman, "Implementing Data Cubes Efficiently" ,Proceedings of the 1996 ACM SIGMOD international conference on Management of data, Montreal, Quebec, Canada, page 205-216, 1996.

[IBM01] IBM, "Nucleus Proves Itself a Super-Scalable iSeries Solution at the Teraplex", IBM Corporation, USA, 2001.

[IntelPentiumD05] Intel, "Intel Pentium D Processor", Intel Corporation, USA, 2005.

[Koller96] Daphne Koller, Mehran Sahami, "Toward Optimal Feature Selection", International Conference on Machine Learning, 1996.

[Li99] Chung-Sheng Li, Philip S. Yu, Vittorio Castelli, "Scan: A Hierarchical Algorithm for Similarity Search in Databases Consisting of Long Sequences, Knowledge Inf. Syst., Volume 1, issue 2, page 229-256, 1999.

[Mumick97] Inderpal Singh Mumick, Dallan Quass, Barinderpal Singh Mumick, " Maintenance of Data Cubes and Summary Tables in a Warehouse " ,1997.

[Pappa04] GL. Pappa, AA Freitas, CAA Kaestner, "Multi-objective algorithms for attribute selection in data mining", *Applications of Multi-Objective Evolutionary Algorithms*, C.A. Coello and G.B. Lamont editors, pages 603-626, December 2004.

[Pasquier00] Nicolas Pasquier, "Data mining: Algorithmes d'extraction et de réduction des règles d'association dans les bases de données" Thèse de doctorat, École Doctorale Sciences pour l'Ingénieur de Clermont-Ferrand, Université de Clermont-Ferrand II, Janvier 2000.

[Provost99] Foster Provost, Venkateswarlu Kolluri, "A Survey of Methods for Scaling Up Inductive Algorithms", Data Mining and Knowledge, Volume 3, Issue 2 , page 131-169, Juin 1999.

[Razente04] Humberto Luiz Razente, Fabio Jun Takada Chino, Maria Camila Nardini Barioni, Agma J. M. Traina, Caetano Traina Jr., "Visual Analysis of Feature Selection for Data Mining Processes", SBBD , page 33-47, 2004.

[Ross97] Kenneth A. Ross, Divesh Srivastava, "Fast Computation of Sparse Datacubes", Proc. 23rd Int. Conf. Very Large Data Bases, VLDB, 1997.

[Salzberg95] Steven Salzberg, Arthur Delcher, David Heath, Simon Kasif, " Best-Case Results for Nearest Neighbor Learning", IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995.

[Sand00] Sand Technology, IBM, "Rapid Development and Deployment of Effective business Intelligence Environments", Sand Technology &IBM Corporation, 2000.

[Schetinin96] Vitaly Schetinin, "Self-Organization of Neuron Collective of Optimal Complexity", Proceedings of Int. Symposium NOLTA'96, Kochi, Japan, page 245-248, 1996.

[Simoudis96] Evangelos Simoudis "Reality Check for Data Mining", IEEE Expert: Intelligent Systems and Their Applications, Volume 11, Issue 5, page 26-33, octobre 1996.

[Tay02] Francis Tay, Lixiang Shen, "A Modified Chi2 Algorithm for Discretization" IEEE Transactions on Knowledge and Data Engineering, volume 14, issue 3, page 666-670, mai 2002.

[Tirthankar00] Tirthankar Lahiri, Serge Abiteboul, Jennifer Widom, "Ozone: Integrating Structured and Semistructured Data", Lecture Notes in Computer Science, 2000.

[Widom95] Jennifer Widom, "Research Problems in Data Warehousing", Proc. 4th Int. Conf. on Information and Knowledge Management (CIKM), Novembre 1995.

[Witten00] Ian H. Witten, Eibe Frank "Data mining, practical machine learning tools and techniques with Java implementations", Morgan Kaufmann Publishers, 2000.

[Yan05] Xifeng Yan, Philip S. Yu, Jiawei Han, "Substructure Similarity Search in Graph Databases", SIGMOD Conference, page 766-777, 2005.