Fouille de données, Contributions Méthodologiques et Applicatives

Martine Collard

Université de Nice – Sophia Antipolis
Sciences et Technologies de l'Information et de la Communication

Mémoire d'Habilitation à diriger des recherches
Informatique

Fouille de données,
Contributions Méthodologiques et Applicatives

soutenu par Martine Collard-Poulard
le 16 décembre 2003

devant le jury composé par

E. Kounalis, Professeur à l'université de Nice-Sophia Antiplois — Président
A. Cavarero-Authosserre, Professeur à l'université de Nice-Sophia Antiplois
J-L. Cavarero, Professeur à l'université de Nice-Sophia Antiplois

Examinateurs

K. Dittrich, Professeur à l'université de Zurich
D. Hérin, Professeur à l'université de Montpellier
M. Tomassini, Professeur à l'université de Lausanne
D.A. Zighed, Professeur à l'université Lumière Lyon2

Rapporteurs

Contenu

Chapitre I Introduction

I.1. Bases de données et Intelligence artificielle

Depuis la définition du modèle relationnel par Ted Codd dans les années 1970, le domaine des bases de données a largement évolué, mais reste encore actuellement largement lié au "relationnel".
Les développements les plus marquants ont été apportés par les approches orientées objet d'une part, et par l'intégration de techniques issues du domaine de l'intelligence artificielle dans les bases de données traditionnelles, d'autre part.
Introduites dans les années 80, les bases de données orientées objet ont intégré des capacités de structuration et d'abstraction puissantes empruntées à la programmation objet avec l'objectif de dépasser les limitations imposées par la simplicité des structures relationnelles.
Alors que les travaux en bases de données orientées objet se focalisent sur la création d'environnement homogènes basés sur un couplage fort entre gestion des données et langages de programmation, le

concept de *bases de données déductives* dote les bases de données des capacités déclaratives des langages logiques.

Une base de données déductive combine des éléments de logique formelle avec une structure relationnelle. Ces bases, encore appelées *bases de données logiques*, définissent des règles exprimées déclarativement sous la forme de clauses de Horn. Les règles sont utilisées pour inférer de nouvelles informations à partir des données (faits) stockées dans la base. Une base de données déductives est ainsi un système expert dont la base de connaissances est constituée par les données relationnelles.

Cependant, les systèmes experts ont déçu, l'expertise faisant souvent défaut et les bases de données déductives ont débouché sur de rares applications ; elles ont fourni pour l'essentiel des résultats théoriques. On peut d'ailleurs citer à ce sujet Jeffrey Ullmann en 1991 : "*Why are OO people writing systems while DD people are writing papers?"* .

Plus récemment, le concept de *base de données intelligente* a été utilisé pour représenter une base de données apte à résoudre des tâches cognitives difficiles. Parsaye et al. [PARS89] voient dans ce concept, *l'évolution et l'intégration de nouvelles technologies incluant l'extraction automatique, l'orientation objet, l'hypermedia, les systèmes experts et les bases de données traditionnelles*. En effet, l'abondance croissante des données stockées dans les organisations, a naturellement amené la communauté des bases de données vers les techniques d'analyse de données et d'apprentissage automatique. Le domaine de la *fouille de données*, que l'on désigne également par *extraction automatique de connaissances à partir de données*, connaît un essor important depuis le début des années 90.

I.2. Fouille de données

La fouille de données, ou *Data Mining* pour les anglo-saxons, était définie en 1996 comme "*a new generation of computational theories and tools to assist humans in extracting useful information (knowledge) from the rapidly growing volumes of digital data"* par Usama Fayyad, Gregory Piatetsky-Shapiro et Padhraic Smyth [FAYY96]

Au travers des multiples tentatives pour caractériser ce domaine, on peut retenir quatre objectifs fondamentaux qui justifient la métaphore de l'extraction et de la transformation de minerai :

- fouiller, creuser, extraire ce qui est caché
- prendre en compte le volume de données

- transformer des données brutes en connaissances expertes
- fournir des connaissances précieuses car nouvelles, valides et utiles à un utilisateur expert.

On utilise fréquemment le terme de *processus de fouille de données* et il s'agit en effet de combiner différentes techniques issues de disciplines diverses dont l'analyse de données, l'intelligence artificielle et l'apprentissage automatique.

La fouille de données concerne des données d'observation par opposition à des données expérimentales qui peuvent être utilisées dans des domaines connexes comme la statistique. En effet, la fouille de données est typiquement dirigée pour l'analyse de données qui ont été préalablement collectées pour d'autres besoins que celui de les analyser. Ceci est un des caractéristiques qui différencie ce domaine de celui de la statistique.

Une autre caractéristique de ce domaine réside dans le volume des données à traiter qui crée des problèmes spécifiques non seulement en termes de performances des systèmes, mais également dans la fiabilité des connaissances découvertes. En quoi une apparente relation peut-elle se généraliser à l'ensemble des données?

Un processus type peut être décrit grossièrement en trois étapes successives de préparation des données, de découverte proprement dite de phénomènes fréquents par la fouille des données et, enfin, de mise en forme des résultats. Les informations extraites lors de la phase de fouille sont en général appelées *modèles* ou *motifs*.

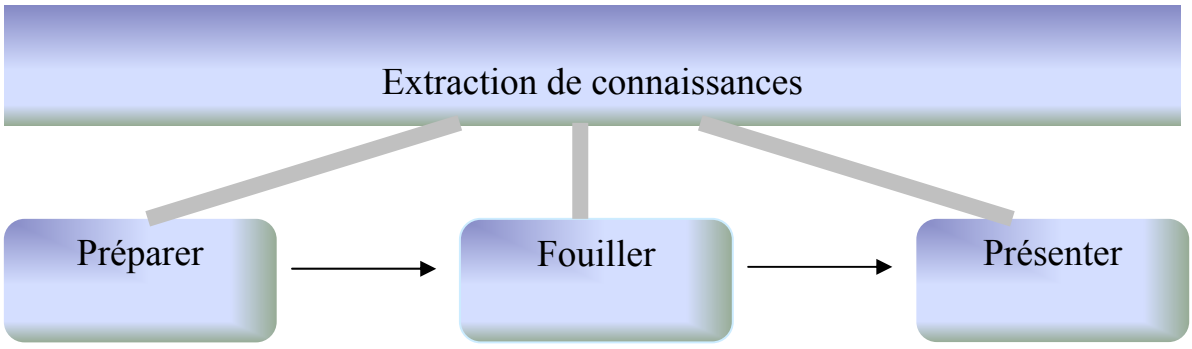


Figure I-1 Les trois phases du processus d'extraction de connaissances à partir des données

La phase de préparation des données passe par les opérations nécessaires de nettoyage (bruit, incohérences), d'intégration cohérente de données issues de sources multiples, de normalisation et de réduction par agrégation ou élimination. Ensuite, intervient la fouille de données, proprement dite, qui consiste à mettre en œuvre différentes tâches permettant l'analyse exploratoire ou bien la recherche de modèles et de motifs. Enfin, la présentation des résultats fait appel à des techniques de représentation des connaissances et de visualisation.

Les travaux présentés dans ce mémoire, ont été développés sur le thème de la découverte de motifs intéressants à travers la fouille de données et mis en œuvre dans le cadre de la conception de systèmes d'information. Ils sont essentiellement consacrés aux problèmes soulevés par l'étape de fouille pour la découverte de modèles et de motifs fréquents. Ils sont à la fois d'ordre méthodologique et applicatif.

I.3. Contribution méthodologique

En ce qui concerne l'aspect méthodologique, ces travaux se situent dans la perspective des problèmes d'optimisation. En fouille de données, la connaissance extraite à partir des données est exprimée de manière structurée sous la forme de modèles et de motifs divers. Un algorithme fournit en général plusieurs modèles ou motifs parmi lesquels il cherche à déterminer le ou les *meilleurs*, ceux qui représentent les données au mieux. La tâche de découverte de modèles intéressants parmi un ensemble de modèles potentiels est typiquement un problème de recherche combinatoire et nécessite souvent d'être accomplie par des techniques de recherche heuristiques. Bien que la recherche de modèles en fouille de données soit rarement présentée selon ce point de vue, il apparaît clairement qu'il s'agit d'un processus d'optimisation. Dans ce contexte, il est nécessaire d'identifier et d'étudier l'espace de recherche, les mesures à optimiser ainsi que les méthodes d'optimisation candidates. Nous mettons en œuvre cette approche dans le contexte de l'extraction des règles de dépendances les plus intéressantes à partir d'un jeu de données. Le point de vue de l'optimisation n'est pas fréquemment considéré dans les travaux sur la découverte de règles ; cependant D. Hand et al. [HAND01] notent que "*l'optimisation et la recherche effective soient souvent sous-estimées en fouille de données alors que le succès des projets en dépendent de manière critique*".

Une règle de dépendance exprime une corrélation entre des conjonctions de termes attribut-valeur. Dans les applications de fouille de données, il est assez fréquent de rechercher des modèles à base de règles étant donné que celles-ci sont aisément compréhensibles par un utilisateur final. Un point

essentiel est d'être capable de mesurer la validité, l'utilité et l'intérêt du lien de dépendance entre l'antécédent et le conséquent d'une règle. Les algorithmes standard utilisent des indices simples pour la sélection de règles. Mais pour satisfaire les objectifs spécifiques de la fouille, des indices plus spécifiques ont été proposés. Certains de ces indices sont équivalents, d'autres sont complémentaires. La recherche des meilleures règles doit s'exprimer ainsi en termes d'optimisation des indices de qualité dans l'espace des règles. Dans l'étude d'un ensemble de règles en tant qu'espace de recherche particulier, nous avons du prendre en compte d'une part sa cardinalité importante et d'autre part la diversité des indices de qualité définis sur les règles. L'intérêt de cette approche réside en particulier dans ce dernier point : l'étude des propriétés particulières de l'espace de recherche permet d'induire les algorithmes de recherche de modèles les plus adéquates.

Les caractéristiques de l'espace des règles qui nous intéressent et les limites des algorithmes couramment mis en œuvre orientent notre choix vers les algorithmes évolutionnaires (AE). Contrairement à la majorité des techniques de recherche qui effectuent une recherche locale, un AE maintient une population de points de l'espace qui lui permet d'opérer de manière globale.

Comme le souligne Schwefel [SCHW00], les AE ne doivent pas être utilisés si une méthode traditionnelle résout de façon satisfaisante le problème. Dans [HAND01], les auteurs soulignent l'intérêt de ces algorithmes en fouille de données, car ils effectuent un traitement global des solutions mais ils s'interrogent sur leur supériorité par rapport aux méthodes standard en fouille de données. Les résultats que nous avons obtenus sur les espaces de règles nous conduisent plutôt à conclure que "cela dépend". En effet, les expériences que nous avons menées, tendent à montrer que les AE permettent d'identifier dans certains cas, des solutions ignorées par d'autres méthodes.

Nous avons dans un premier temps conduit des expériences visant à optimiser un seul indice ; puis, nous avons pris en compte simultanément plusieurs indices. Les problèmes d'optimisation multi-critères ont été traités soit par des méthodes scalaires, soit par des méthodes basées sur un critère de Pareto. Nous avons testé une méthode d'optimisation scalaire où la fonction à optimiser est une somme pondérée des indices à considérer simultanément. Puis, nous avons donc recherché une méthode mieux adaptée à l'optimisation multicritères ce qui nous a amené à appliquer l'algorithme NSGA (Non dominated Sorting Genetic Algorithm) [SRIN95] à notre problème.

Ainsi, le problème fondamental et difficile de définition d'indices peut être traité par une technique multi-critères ; à partir d'un ensemble de critères "de base ", il est possible de trouver des règles qui représentent des compromis satisfaisants. Cette approche fournit un outil d'aide à la décision dans le choix de règles pertinentes pour l'utilisateur.

I.4. Contribution applicative

En parallèle avec nos recherches sur le thème de l'optimisation (cf. Figure) et en ce qui concerne l'aspect applicatif de nos travaux, nous avons mis en œuvre des techniques de fouille de données sur des problèmes classiques de *credit scoring* et de téléphonie mobile. Deux nouveaux projets vont concerner la fouille de données en bioinformatique avec les puces à ADN, d'une part et une application de type CRM (Customer Relationship Management) d'autre part.

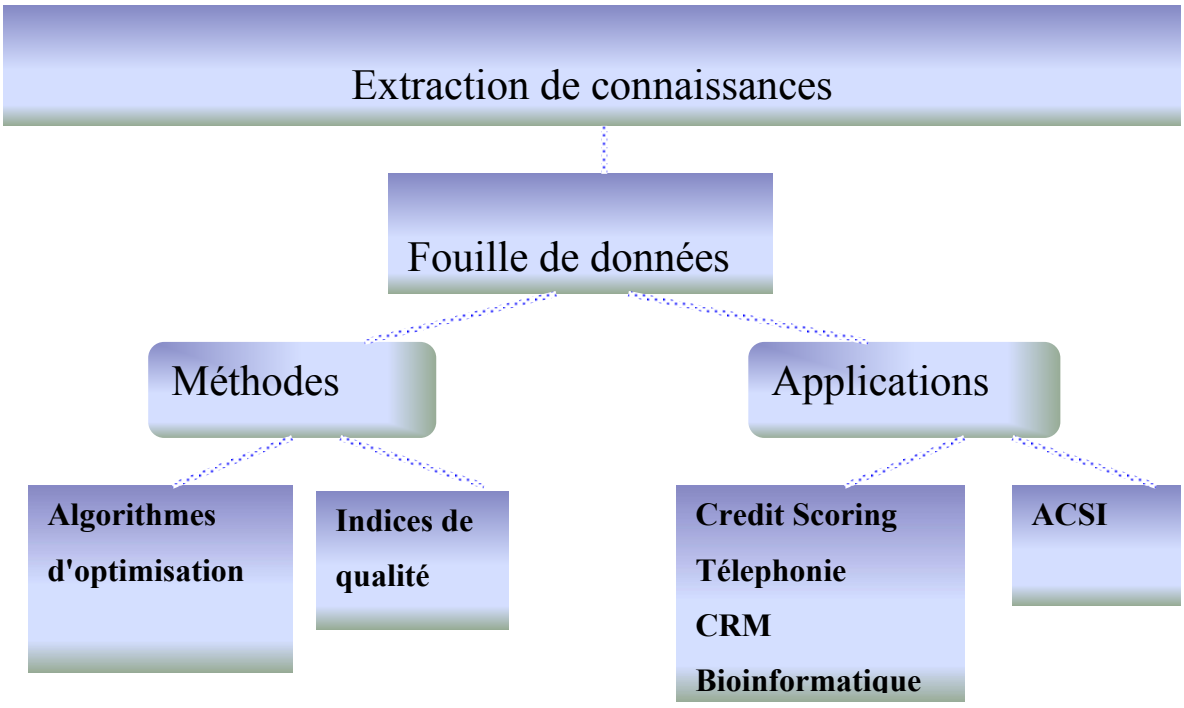


Figure I-2 Organisation de nos activités de recherche en fouille de données

C'est dans le cadre du projet MECOSI (MEthodes de Conception des Systèmes d'Information) au laboratoire I3S et sur le thème de la ré-ingénierie des systèmes d'information, que se situe notre contribution majeure en termes d'application. L'application de techniques de fouille de données pour l'ingénierie des systèmes d'information n'est pas courante bien qu'elles apportent des solutions pour

traiter un problème de ré-ingénierie. Nous avons travaillé sur ce sujet dans le cadre de la rétro-conception de bases de données. Un processus de rétro-conception de bases de données consiste en l'application de méthodes permettant d'automatiser la compréhension de la structure et de la sémantique des données et la construction d'un nouveau schéma conceptuel. Nous nous sommes placés dans le contexte des bases de données relationnelles qui sont actuellement les plus utilisées.

Nous avons proposé la méthode EXORE (EXtraction de schémas Objet pour la REtro-conception de bases de données relationnelles) qui répond à différents objectifs motivés par l'observation des bases de données actuelles et des méthodes proposées. Une première motivation a été de définir une approche réaliste pour des bases de données relationnelles en usage, éventuellement dégradées. Un deuxième objectif a été d'exploiter le volume très important des informations de toute sorte, stockées dans ces bases anciennes, par des techniques appropriées issues du domaine de la fouille de données. Nous avons, en particulier défini une distance contextuelle permettant d'extraire des similarités entre attributs à partir d'un ensemble de données volumineux issu de requêtes posées par les utilisateurs. Une originalité de l'approche est de considérer une base de requêtes fournie par les utilisateurs comme une base de données et d'y appliquer des techniques de fouille.

La définition d'un modèle pour la description du schéma cible du processus de rétro-conception constitue un autre sujet d'importance, car il peut faciliter l'implémentation future du schéma. Nous avons défini, un modèle logique spécifique, proche des modèles actuellement utilisés : relationnel, objet et relationnel-objet.

La méthode EXORE se caractérise également par l'application de deux phases successives : une phase d'analyse et de découverte, suivie d'une phase de construction d'un schéma abstrait. La phase d'analyse permet non seulement d'établir un catalogue des structures et informations explicitement déclarées, mais également de mettre à jour l'information implicitement stockée dans la base. La découverte des anomalies de nommage, du non-respect des contraintes d'intégrité, de structures optimisées intervient à ce niveau et alimente la phase de construction qui peut ensuite produire le schéma résultant. La phase de construction utilise, en particulier, les similarités identifiées et applique des procédures permettant de détecter les entités-classes, les entités-relations et les liens éventuels entre entités en minimisant les interactions avec les experts.

I.5. Plan du mémoire

La suite du mémoire est organisée en trois chapitres :

- le chapitre II présente l'aspect méthodologique de nos travaux qui considèrent la fouille de données selon la perspective d'un problème d'optimisation,
- le chapitre III présente l'aspect applicatif de nos travaux et se focalise sur le problème de la retro-conception de bases de données et la mise en œuvre de techniques de fouille de données,
- le chapitre IV présente les projets de recherche en cours et conclut.

Chapitre II La fouille de données : un problème d'optimisation

---

II.1. Introduction

Nous avons cité, en introduisant le domaine de la fouille de données, la définition proposée par Usama Fayyad, Gregory Piatetsky-Shapiro et Padhraic Smyth en 1996. En quelques années, le domaine a très largement élargi l'étendue de ses champs d'intérêt et la manière d'appréhender le sujet

évolue rapidement. Citons par exemple, la définition que l'on peut trouver actuellement sur le site de l'équipe IA du laboratoire LRI (Paris 6) : *"son but est de lutter contre le morcellement des expertises et la dilution des connaissances, et de restituer à l'expert humain la compréhension et la maîtrise des phénomènes latents dans les monceaux de données".*

Restituer la compréhension est une tâche plus ambitieuse que la recherche de modèles et ouvre de nouveaux horizons de recherche. Par exemple, la manière la plus classique d'appréhender un problème de fouille de données consiste à sélectionner les tâches de modélisation et les techniques employées selon les objectifs de la personne qui analyse les données.

On distingue essentiellement deux sortes de tâches: description ou prédiction. Dans chacun des cas, les techniques sont choisies en fonction du type et de la structure du résultat souhaité ; des mécanismes d'optimisation sont implicitement mis en œuvre pour sélectionner les meilleurs parmi l'ensemble des résultats possibles. Il n'est pas apparu pour l'instant de techniques dont les performances soient les meilleures quelque soit le problème à analyser ; aussi, l'analyste procède fréquemment par un jeu d'essais et erreurs en appliquant différentes techniques disponibles. Par exemple, la recherche d'un classifieur caractérisant au mieux les données pourra aussi bien être réalisée en appliquant un algorithme d'induction d'arbre, un réseau de neurones ou encore une classification bayésienne sans que l'on puisse prédire de la supériorité d'une technique.

Ce constat nous amène à suivre une approche différente, qui n'exclut pas les précédentes mais les complète, qui n'est pas dirigée par les tâches mais qui considère la fouille de données selon le point de vue d'un processus d'optimisation. Ceci implique d'étudier l'espace de recherche, les indices à optimiser et les méthodes d'optimisation candidates. L'espace de recherche est celui des règles de dépendance de la forme $A \rightarrow B$ où A et B sont des conjonctions de termes attribut-valeur $Att\ op\ a$, *Att* est un attribut, a est une de ses valeurs et $op \in \{=, <, >, \leq, \geq\}$ et pour lesquelles il s'agit d'optimiser la validité, l'utilité et l'intérêt du lien de causalité entre l'antécédent et le conséquent.

En fouille de données, on retrouve souvent les principes d'*intelligibilité* et d'*intéressabilité* des connaissances extraites. Le caractère *intéressant* d'une connaissance étant éminemment flou et informel, les indices tentant de quantifier ce critère revêtent des aspects différents, mais visent en général à évaluer la validité, l'utilité et l'intérêt pour l'utilisateur du lien de dépendance entre l'antécédent et le conséquent des règles.

Les algorithmes issus du domaine de l'apprentissage automatique, recherchent des règles générales c'est-à-dire couvrant de nombreuses données plutôt que des règles plus spécifiques. Ils visent à extraire un ensemble de règles qui forment un modèle supposé s'appliquer à toutes les données. Aussi,

la qualité d'un modèle extrait est évaluée selon sa *précision* (ou *taux de succès*) qui représente le taux de prédictions correctes que l'ensemble tout entier des règles réalise sur des exemples non connus précédemment.

En revanche, en fouille de données, les règles sont plutôt considérées de façon individuelle en dehors du modèle qui les compose. Le but est souvent de découvrir des motifs intéressants qui ne sont pas seulement valides mais également nouveaux, non triviaux ou surprenants. Contrairement aux modèles, ces *motifs* intéressants peuvent ne pas couvrir de larges ensembles de données. Les critères pour l'évaluation de la qualité sont mesurés sur une règle considérée individuellement plutôt que collectivement.

Nous suivons une approche similaire en considérant les règles de manière générale comme des liens de dépendance et nous étudions l'intérêt et l'utilité de règles prises individuellement. L'étude de l'espace de recherche et des indices à optimiser nous a conduit à observer la densité d'état de l'espace de règles, c'est à dire la distribution des valeurs pour un indice donné, et d'autre part, le comportement relatif de certains indices pris deux à deux. Nous avons ainsi mis en évidence la diversité des densités d'état parmi les différents jeux de données testés d'une part et des antagonismes entre indices de qualité sur certains jeux de données d'autre part. Alors que les méthodes d'optimisation généralement utilisées en fouille de données sont des méthodes locales, les travaux utilisant des meta-heuristiques sont rares bien que celles-ci soient mieux adaptées aux espaces de recherche induits. Parmi les différentes meta-heuristiques, les algorithmes évolutionnaires (AE) exploitent, dans un contexte d'ingénierie, le principe de la sélection naturelle des espèces vivantes. Ils présentent des avantages particuliers pour être appliqués à la problématique de la fouille de données. En particulier, ces algorithmes manipulent et évaluent chaque solution (règle) dans sa globalité contrairement aux méthodes évaluant un terme attribut-valeur à la fois. Cette qualité leur confère une plus grande efficacité dans la recherche parmi des données très corrélées. D'autre part, le fait de travailler sur un ensemble de solutions améliorées progressivement, plutôt que de n'en considérer qu'une seule à la fois est particulièrement adapté aux problèmes multi-critères dans lesquels des compromis doivent être trouvés. Les résultats que nous avons obtenus sur les espaces de règles nous conduisent, en effet, à conclure que les AE permettent d'identifier dans certains cas, des solutions ignorées par d'autres méthodes.

La suite du chapitre est organisée de la manière suivante :

- la section II.2 présente les tâches les plus courantes en fouille de données et les techniques d'optimisation mises en œuvre,

- dans la section II.3, selon le point de vue "optimisation" que nous avons choisi, nous étudions les caractéristiques des espaces de règles,
- la section II.4 est consacrée aux algorithmes de recherche adaptés aux espaces de règles et nous centrons l'étude sur les algorithmes génétiques,
- enfin, nous concluons et présentons nos projets de travaux dans la section II.5

II.2. Fouille de données et Optimisation

II.2.1. Modèles et Motifs

La connaissance extraite par des techniques de fouille de données est représentée sous des formes diverses ; une première distinction qui peut être faite concerne la couverture de l'espace des données. On parle généralement de *modèle* pour désigner une structure informationnelle globale qui caractérise l'ensemble des données. Par opposition, on utilise le terme de *motif* (pattern) pour traduire la localité de certaines descriptions qui recouvrent une région limitée de l'espace. Cette distinction est fondamentale en fouille de données où l'objectif est d'identifier des phénomènes inattendus qui peuvent ne caractériser qu'une partie, éventuellement peu étendue, des données. La rareté d'un motif peut d'ailleurs contribuer à son intérêt.
On présente aussi la fouille de données en se référant aux objectifs de l'utilisateur. On distingue ainsi les tâches de description qui donne une vue synthétique des données pour améliorer leur compréhension, et les tâches de prédiction qui cherchent à prédire les valeurs d'une variable *cible* à partir des valeurs d'autres variables dites *prédictives*.

II.2.2. Tâches

Dans ce paragraphe, nous présentons les tâches les plus fréquentes en fouille de données, qui illustrent à la fois l'aspect local ou global des modèles et l'objectif de prédiction ou de description.
La fouille de données débute en général par une phase d'analyse exploratoire qui permet d'avoir une première vision des données. Cela peut consister, par exemple, à appliquer une analyse en composantes principales ou des méthodes de représentation graphique. Ensuite, la recherche de groupes de données similaires constitue en général, une phase centrale en fouille de données.

- **Classification non supervisée**

Un modèle descriptif vise en général à décrire la globalité des données. On peut ranger dans cette catégorie l'estimation de densité ainsi que la *segmentation* et la *classification non supervisée*.

En *classification non supervisée*, le but est de partitionner l'ensemble des données en un certain nombre de *classes* homogènes. Cela signifie que les classes doivent contenir des données qui partagent un haut degré de similarité ; le but est de maximiser la similarité à l'intérieur d'une classe et de minimiser la similarité entre classes. La similarité des objets est mesurée en termes de distance entre les données. La difficulté réside dans la recherche de modèles qui optimisent non seulement les distances intra-classe et inter-classes, mais également le nombre de classes qui doit rester relativement faible.

En fouille de données, l'utilisateur n'a en général, aucune connaissance a priori des données et ne peut pas connaître le nombre idéal de classes ; aussi, on utilise plutôt des algorithmes qui déterminent automatiquement ce nombre. Les algorithmes de classification sont particulièrement nombreux ; sont proposées soit de nouvelles variantes pour d'anciennes méthodes, soit de nouvelles approches. Il est habituel de distinguer les méthodes de partitionnement et les méthodes hiérarchiques, bien que cette dichotomie apparaisse moins pertinente sur certaines méthodes récentes.

L'étape suivant la classification consiste à poursuivre l'analyse en recherchant comment prédire l'appartenance d'une nouvelle donnée à une classe.

- **Modélisation prédictive**

La modélisation prédictive peut être vue comme la recherche d'une fonction qui associe une classe à une donnée à partir de la valeur des variables qui la décrivent. La classe est représentée par une valeur devariable appelée *variable cible* ou *variable de classe*. Par exemple, un modèle prédictif va permettre de prédire les risques associés à un accord de crédit bancaire, la résiliation d'un contrat par un client en téléphonie mobile ou encore, la fraude dans une déclaration d'accident à une compagnie d'assurance. On utilise le terme de *classification supervisée* lorsque la variable de classe est catégorielle et le terme de *régression* si la variable est réelle.

Le but est de construire un modèle permettant de classer toute donnée nouvelle, inconnue pendant la phase de construction du modèle. Les travaux sur la modélisation prédictive, comme pour la classification non supervisée, ne sont pas spécifiques du domaine de la fouille de données ; dans le cadre de l'apprentissage automatique, on peut citer de nombreuses propositions sur les algorithmes de classification [GELF91, ZIGH98, KARY99, WANG99]. Quant à la question du choix de la meilleure méthode entre un réseau de neurones, une méthode d'induction d'arbres de décision, un classifieur

Bayésien …, il n'y a pas de réponse simple, car cela dépend des objectifs, des données et des caractéristiques du problème.

Nous citons ici les méthodes d'induction d'arbres de décision comme référence, car ces structures se traduisent simplement sous forme de règles et intègrent aussi bien des variables prédictives discrètes que continues. La construction de l'arbre de décision consiste à utiliser les variables prédictives pour partitionner récursivement l'ensemble des données en sous-ensembles de manière à maximiser la pureté de la partition. Cette technique nécessite de déterminer à chaque étape les attributs les plus discriminants, c'est à dire les plus significatifs pour le partitionnement.

On peut voir l'arbre de décision comme un enchaînement hiérarchique de propositions logiques construites de manière automatique à partir d'une base d'exemples. Un exemple est constitué d'une liste d'attributs, dont la valeur détermine l'appartenance à une classe donnée. Il existe une grande variété d'algorithmes pour construire des arbres de décision, on peut citer par exemple CART (Classification And Regression Trees), CHAID (CHi-squared Automatic Interaction Detection) ou C4.5 [QUIN96]. L'algorithme CART [BREI84] construit un arbre binaire en divisant les enregistrements au niveau de chaque nœud selon une fonction ne dépendant que du champ d'entrée, l'arbre est ensuite élagué pour réduire le taux de mauvaises classifications sur les nouveaux exemples. CHAID [HART75] essaie d'arrêter la croissance de l'arbre avant la construction de branches qui seront ensuite élaguées. C4.5 est un des algorithmes les plus connus et crée un nombre variable de branches par nœud. Dans l'étude qui est décrite ci-après, nous utilisons l'implémentation de l'algorithme C4.5, nommée J48 disponible dans le système Weka[1].

C4.5 met en œuvre le principe "diviser pour régner" qui consiste à choisir, à chaque itération, une variable qui représente un nœud de l'arbre dont partent les branches étiquettées par des valeurs de la variable. Le choix de la variable est basé sur un critère mesurant le pouvoir discriminant de la variable ; le *gain d'information* est calculé à partir de l'entropie. L'arbre est construit nœud après nœud, en considérant un attribut à chaque étape, jusqu'à épuisement des attributs ou des données à classer. Chaque feuille représente une classe (valeur de la variable de classe). La construction de l'arbre est réalisée en utilisant une partie du jeu de données que l'on appelle *l'ensemble d'apprentissage*. L'arbre ainsi construit est ensuite élagué. L'élagage permet de produire des modèles plus courts et donc, plus compréhensibles, et, également, de réduire le taux de mauvais classements de nouvelles données issus de *l'ensemble de test* qui est distinct de *l'ensemble d'apprentissage.*

[1] Logiciel de fouille de données développé à l'université de Waikato http://www.cs.waikato.ac.nz/ml/weka/

- **Recherche d'associations**

La classification non supervisée et le classement produisent des *modèles* globaux qui se décrivent la totalité de l'espace des données, alors que la recherche d'associations est typiquement issue des applications de fouille de données dans lesquelles on recherche plutôt des *motifs* locaux. On cherche, par exemple, à caractériser des comportements atypiques en détection de fraudes, ou bien encore à décrire 10% des clients qui sont susceptibles de répondre à un mailing.

Une règle d'association est définie syntaxiquement comme une règle logique d'ordre 0 de la forme *si A alors B* où les propositions A et B sont des conjonctions d'expressions simples de comparaison sur les attributs. Cependant, la sémantique qui lui est associée ne correspond pas à la formule logique *nonA ou B*. Hand et al. [HAND01] soulignent d'ailleurs l'inadéquation du terme de *règle* pour ces motifs qui ne sont pas sélectionnés sur le lien de causalité qu'ils sont censés refléter.

Les premiers travaux sur ce sujet ont été introduits par R. Agrawal et al. [AGRA93] et ont suscité de nombreuses recherches. Ils ont été développés à l'origine pour l'analyse de bases de données de transactions de ventes, et avaient pour objectif de découvrir des relations significatives entre les produits vendus. Ils attribuent à ces règles, une nature probabiliste dans l'indice où une telle règle n'a de signification que si elle est associée à son indice de *confiance* qui représente la probabilité que son conséquent soit réalisé sachant que son antécédent est réalisé. Ces règles particulièrement adaptées à des attributs booléens (ou plus généralement discrets), ont une forme simple et interprétable qui explique leur succès :

$$si\ att_1=1\ ET\ att_5=1\ ET\ \ldots\ alors\ att_8=1\ ET\ att_4=1\ avec\ la\ probabilité\ p$$

Pour une règle d'association, l'indice de *support* qui représente la proportion de données pour lesquelles l'antécédent et le conséquent sont simultanément réalisés est également caractéristique ; il est le premier critère de sélection sur lequel sont fondés de nombreux algorithmes de recherche d'associations. Le choix du seuil min_{supp} de *support* minimum oriente la recherche vers des règles plus ou moins générales. Aussi, dans certains cas, les motifs ainsi extraits ne recèlent aucune information pertinente.

La cardinalité de l'espace de recherche est grande, en $O(p^{2^{p-1}})$ pour p attributs binaires. Les algorithmes utilisent la propriété d'anti-monotonie du support pour réduire leur complexité. En effet, on peut remarquer que

$$support(att_i=1\ ET\ att_j=1) < support(att_i=1)$$

Les algorithmes utilisent cette propriété en recherchant en premier lieu les termes simples *att=1* dont le support est supérieur à min_{supp}. On parle dans ce cas d'*item* fréquent. Un ensemble formé de deux *items* fréquents est candidat à être fréquent, mais si *I1* est un *item* fréquent et *I2* n'est pas fréquent, l'algorithme n'étudie pas *I1 ET I2*. Ce principe est appliqué pour identifier un ensemble fréquent de k *items* à partir des ensembles fréquents de k-1 *items*, en éliminant tout ensemble de k *items* dont au moins un sous-ensemble de k-1 *items* n'est pas fréquent. Parmi la liste des ensembles de k *items* candidats, l'algorithme parcourt le jeu de données pour déterminer lesquels sont effectivement fréquents. Les ensembles de k *items* qui sont conservés sont, à leur tour, combinés pour former les ensembles candidats de k+1 *items*. Ce processus est itéré jusqu'à ce qu'aucun nouvel *item* ne puisse être généré. Finalement un parcours des données est effectué pour sélectionner, parmi les ensembles fréquents, ceux qui représentent une règle de confiance supérieure au seuil minimum.
Cet algorithme d'extraction est connu sous le nom d'*APriori*. Il a été introduit par Agrawal et Srikant [AGRA94]. De nombreuses variantes de ce schéma de base ont été proposées pour optimiser le parcours des données, le nombre d'ensembles d'*items* candidats ou le temps de calcul des valeurs de *support* et de *confiance*. Notons que l'algorithme décrit ci-dessus s'applique de la même façon à des règles plus générales dans lesquelles les items sont des termes plus complexes portant sur des variables non booléennes.

- **Motifs locaux**

Parmi les motifs, on distingue une classe de modèles très locaux que l'on désigne par le terme d'*évènements rares* ou de *règles locales* (*small disjuncts*). Les travaux consacrés à ses motifs particuliers restent pour l'instant marginaux. Holte et al. [HOLT89] ont, les premiers, décrit le problème des évènements rares définis comme des règles couvrant peu d'exemples et sur lesquelles le taux d'erreur d'un classifieur se concentre. Ils ont montré que l'élimination de ces motifs n'affecte pas le taux d'erreur car les exemples qu'ils couvraient sont alors mal classés. Weiss [WEISS98] a recherché les interactions entre les évènements rares et le bruit ; il a montré que l'introduction de données bruitées augmente le taux d'erreur sur les règles très locales. Dans [WEISS00], une étude expérimentale menée sur 30 jeux de données a montré que le taux d'erreur n'est pas toujours concentré sur les règles locales. Cette étude a permis d'explorer les liens entre élagage et règles locales et a montré que l'élagage n'a pas d'effet lorsque le taux d'erreur du modèle n'est pas concentré sur les règles locales.

II.2.3. Recherche des meilleurs modèles

Nous avons pu voir que la connaissance extraite à partir des données est exprimée de manière structurée sous la forme de modèles et de motifs divers. Un algorithme fournit en général plusieurs modèles parmi lesquels il cherche à déterminer le ou les *meilleurs*, ceux qui représentent les données au mieux. Ceci est réalisé par une fonction de score applicable à chaque modèle et l'optimisation consiste à rechercher un optimum pour cette fonction. David Hand et al. [HAND01] soulignent le fait que "*l'optimisation et la recherche effective soient souvent sous-estimées en fouille de données alors que le succès des projets en dépendent de manière critique*".

La tâche de découverte de modèles intéressants parmi un ensemble de modèles potentiels est typiquement un problème de recherche combinatoire et nécessite souvent d'être accomplie par l'application d'heuristiques de recherche. En régression linéaire, une règle de prédiction est souvent trouvée en minimisant la fonction des moindres carrés, somme des carrés des erreurs produites par un modèle par rapport aux données observées. Dans ce cas, des méthodes algébriques permettent de trouver le modèle optimal. Par contre, dans le cas de l'apprentissage supervisé, il est plus difficile de minimiser analytiquement le taux d'erreur d'un classifieur ; la recherche du meilleur classifieur sous forme d'un arbre de décision ne peut être effectuée par un parcours exhaustif de l'espace des solutions du fait de la combinatoire du problème. En fouille de données, l'explosion combinatoire du nombre des modèles possibles est courante. C'est le cas, par exemple, dans la recherche des meilleurs ensembles de conditions dans la partie gauche d'une règle. En général, à moins que la fonction de score ne soit très simple ou que l'espace des règles soit petit, il est difficile de proposer des techniques d'optimisation qui garantissent de trouver un optimum global car le parcours exhaustif de l'espace de recherche est impossible. Les techniques d'optimisation courantes comme la méthode du gradient ne s'appliquent pas dans ce domaine où le type de problème est discret.

Un problème d'optimisation combinatoire est un problème où il faut minimiser une certaine fonction (dite de coût) sur un ensemble fini de configurations. Ce type de problème apparaît très souvent dans l'organisation des activités humaines. Par exemple créer les emplois du temps des élèves et des professeurs dans une école, organiser les étapes de production dans une entreprise, choisir le trajet des facteurs. La résolution de ces problèmes combinatoires repose sur des méthodes de recherche heuristiques. Parmi ces méthodes, certaines sont connues pour donner en moyenne de bonnes

performances, elles produisent des solutions presque optimales assez rapidement et dans de nombreux cas, elles localisent un optimum local. Les méthodes de recherche locales, par exemple, sont ainsi nommées parce qu'à partir d'un modèle M, elles cherchent un autre modèle parmi ceux qui sont dans son voisinage. Les méthodes les plus fréquemment employées commencent par le choix d'un point dans l'espace de recherche ; le choix peut être aléatoire ou dirigé. La recherche procède ensuite en opérant à chaque itération, un déplacement par rapport au point courant en modifiant celui-ci. La qualité du nouveau point est évaluée et la décision de remplacer le point courant par le nouveau point est prise si celui-ci est meilleur. Ces techniques qualifiées de *recherche gloutonne*, reposent le plus souvent sur l'itération d'étapes de recherche locale dans le voisinage de l'état courant. La qualité de la solution produite est ainsi dépendante du point de départ de l'algorithme. Ces méthodes conduisent le plus souvent à trouver uniquement un optimum local.

Par exemple, l'induction d'un arbre de décision par un algorithme comme C4.5 met en œuvre ce type de recherche gloutonne. Après avoir choisi une fonction d'évaluation adéquate, le *gain d'information*, par exemple, ces algorithmes suivent souvent une méthode heuristique du type "général vers spécifique" : l'algorithme débute avec la forme (règle) la plus générale possible, puis spécialise cette forme à chaque étape, explorant ainsi le voisinage de la règle.

En revanche, avec un algorithme d'extraction d'associations de type APriori, il s'agit d'une recherche systématique dans l'espace en profondeur suivie d'une phase d'élagage. La fonction d'évaluation est binaire : une règle a la valeur 1 si et seulement si son indice de *support* et de *confiance* sont tous deux supérieurs à leur seuil minimum respectif. La taille de l'espace de recherche croît là encore exponentiellement avec le nombre d'attributs ; APriori optimise le parcours de l'espace en exploitant les caractéristiques des indices et en particulier, l'anti-monotonie du *support*. Alors que les algorithmes d'extraction de règles existaient déjà en nombre en apprentissage automatique, les algorithmes de règles d'associations sont typiquement adaptés pour opérer de manière efficace, sur des jeux de données volumineux. D'ailleurs, les recherches sur ces algorithmes se focalisent plus sur les performances d'exécution que sur l'interprétation de la règle produite. Or, les règles sélectionnées sur des critères aussi simples sur le *support* et la *confiance* sont généralement très nombreuses et nécessitent des phases ultérieures de sélection.

II.3. Etude de l'espace de recherche

Nous l'avons vu précédemment, la tâche de découverte de motifs intéressants parmi un ensemble de modèles potentiels est un problème de recherche combinatoire. Pour formuler la problématique de fouille de données comme un problème d'optimisation, il est nécessaire de traduire les objectifs sous une forme exploitable. Il s'agit donc d'identifier et d'étudier l'espace de recherche, les indices à optimiser et les méthodes d'optimisation candidates. L'étude des propriétés particulières de l'espace de recherche permet ainsi de déterminer les algorithmes de recherche de modèles les plus adéquats.

Les règles de dépendance de la forme $A \rightarrow B$ que nous considérons sont évaluées en fonction d'indices statistiques traduisant l'intérêt du lien de causalité entre A et B. Pour étudier les propriétés des ensembles de règles considérés comme espaces de recherche particuliers, nous avons du prendre en compte d'une part une cardinalité importante et d'autre part la diversité des indices de qualité définis sur les règles. L'étude de l'espace des règles ne peut pas être réalisé par un parcours exhaustif et impose de choisir un échantillon qui soit le plus représentatif. Nous proposons d'appliquer une méthode d'échantillonnage basée sur une analogie avec la thermo-dynamique.

En ce qui concerne les indices d'intérêt, nous considérons un certain nombre d'indices "de base" évaluant différentes qualités du lien de causalité entre prémisse et conséquent. Nous étudions leur distribution dans l'espace des règles, leur distribution selon la longueur des règles et leur corrélation mutuelle dans cet espace.

Dans la suite de cette section, nous détaillons les propriétés observées expérimentalement après avoir préciser le concept de règle de dépendance et donner un aperçu des indices de qualité proposées pour une règle et des études réalisées sur ce sujet. La section II.4, quant à elle, sera consacrée aux différents algorithmes utilisés.

II.3.1. Règles de dépendances

Parmi les objectifs présentés comme essentiels en fouille de données, on retrouve souvent le principe d'*intelligibilité* des connaissances extraites. Il est ainsi assez fréquent de rechercher des modèles à base de règles étant donné que ceux-ci sont aisément compréhensibles par un utilisateur final. Un autre principe est celui de l'*intéressabilité* des connaissances. Le caractère *intéressant* d'une connaissance étant par nature, flou et informel, les indices tentant de quantifier ce critère revêtent des aspects

différents, mais visent en général à évaluer la validité, l'utilité et l'intérêt pour l'utilisateur du lien de dépendance entre l'antécédent et le conséquent d'une règle.

Les algorithmes standard utilisent des critères simples pour la sélection de règles comme nous l'avons montré au paragraphe II.2. Mais pour satisfaire les objectifs de la fouille, des indices plus spécifiques ont été proposés. La recherche des meilleures règles doit s'exprimer ainsi en termes d'optimisation des indices de qualité dans l'espace des règles. Les critères simples pour l'évaluation de règles comme la *précision*, le *support* et la *confiance* sont maintenant connus pour être insuffisants si l'on veut extraire des informations utiles et intéressantes. Pour la qualité globale d'un classifieur, la *sensibilité* et la *spécificité* fournissent plus d'information sur la qualité du modèle. L'utilité et l'intérêt peuvent avoir différents sens et on distingue des approches objectives et subjectives. Les critères subjectifs sont généralement basés sur une comparaison des règles apprises par rapport à une connaissance a priori sur les données. Un des premiers indices objectifs, *RI (Rule Interest)* a été définie par G. Piatetsky-Shapiro [PIAT91] en 1991. De nombreuses autres indices ont été proposés depuis, pour l'évaluation de la qualité des informations extraites : le *lift*, la *J-Measure*, l'indice définie par Sebag et Schoenauer, la *conviction*. Certains de ces indices sont équivalents, d'autres sont complémentaires.

De nombreux algorithmes ont été proposés pour l'induction de règles à partir de données dans le cadre de l'apprentissage automatisé et sont également utilisés pour la prédiction en fouille de données. Ils fournissent des règles de classification dont la partie droite est prédéfinie et représente la classe (cf. section II.2). Ces méthodes recherchent préférentiellement des règles de prédiction générales c'est-à-dire couvrant de nombreuses données plutôt que des règles plus spécifiques. Elles visent à extraire un ensemble de règles qui forment un classifieur précis et ce modèle est supposé s'appliquer à toutes les données. Ainsi, la qualité d'un modèle extrait est évaluée selon sa *précision* qui représente le taux de classifications correctes que réalise l'ensemble tout entier des règles sur des exemples non connus lors de la phase d'apprentissage. En revanche, en fouille de données, les règles sont plutôt considérées de façon individuelle en dehors du modèle qui les compose. Le but est souvent de découvrir des motifs intéressants qui ne sont pas seulement valides mais également nouveaux, non triviaux ou surprenants. Contrairement aux modèles, ces *motifs* intéressants peuvent ne pas couvrir de larges ensembles de données. Les critères pour l'évaluation de la qualité sont mesurés sur une règle considérée individuellement plutôt que collectivement. Une technique pour générer un ensemble de règles individuellement intéressantes et utiles est de construire un arbre de classification et ensuite d'évaluer chacune des branches comme une règle individuelle selon des critères spécifiques de qualité. Mais certaines des règles produites par ces techniques standard de classification, n'ont pas de sens pour

l'utilisateur puisque ces techniques utilisent des biais et des heuristiques spécifiques pour générer le classifieur. Aussi, comme le montre [DHAR00], avec ce type de procédé, on peut ignorer des règles intéressantes.

Les règles d'association sont des motifs locaux typiques des applications du type "panier de la ménagère" où les relations fréquentes entre les produits sont susceptibles d'être découvertes. Dans ces règles, la cible n'est pas prédéfinie, la partie droite peut être une conjonction de conditions d'attributs ainsi que nous l'avons montré dans la section II.2. Une règle d'association peut être considérée comme une propriété probabiliste relative à la co-occurence d'événements qui satisfont des contraintes statistiques sur la base de données comme un *support* et une *confiance* minimaux. En termes de probabilité, le *support (supp)* de la règle $A \rightarrow B$ représente $p(A \cap B)$ et la *confiance (conf)* représente la probabilité conditionnelle $p(B/A)$ où par abus d'écriture, on note respectivement A et B , l'ensemble des exemples vérifiant l'antécedent A et le conséquent B.

Règles de classification et règles d'association sont deux concepts importants en fouille de données. A. Freitas [FREI98b] met en évidence les différences entre ces deux concepts. Cependant un certain nombre de techniques combinent ces deux modèles. Par exemple, [LIU98] propose d'intégrer les deux types de règles pour construire des classifieurs qui sont plus adaptés aux objectifs de la fouille de données. Cette approche utilise les techniques d'extraction des règles d'association pour la tâche de classification. L'algorithme fondateur *APriori* [AGRA93, AGRA94] est adapté pour cette tâche. Selon cette technique, les règles sont sélectionnées si elles satisfont des seuils minimaux de *support* et *confiance* qui ne sont pas des critères standards pour les règles de classification. Ces règles prédisent une valeur d'attribut de classe et elles sont considérées individuellement. Dans [DHAR00], le système est conçu pour la recherche de règles de classification pour des applications financières. L'évaluation des règles est réalisée selon deux critères : le taux de mauvais classements sur des données inconnues pour le modèle entier, et les performances en termes de support et confiance pour les règles individuelles. Il est en effet indiqué que dans les problèmes financiers, il est peu probable qu'un modèle puisse classer précisément tous les cas. Le système est conçu pour ne trouver éventuellement que peu de règles utiles et très précises. La qualité des règles est déterminée grâce au *support*, à la *précision*, à *l'entropie* ou à certaines combinaisons de ces métriques sur des règles individuelles.

Nous suivons une approche similaire : nous étudions l'intérêt et l'utilité de règles prises individuellement. Nous considérons les règles comme des liens de dépendance. Les règles simples sont caractérisées par le fait que leur conséquent est réduit à un seul terme attribut-valeur. L'opérateur utilisé est généralement l'opérateur d'égalité, cependant il est envisageable d'employer des opérateurs

ensemblistes ou logiques lorsque le domaine de l'attribut s'y prête. Ces règles sont donc de la forme suivante :

$$\text{si } A_1 \; o\text{p } a_1 \text{ET} \ldots \text{ET } A_m \; o\text{p } a_m \;\; a\text{lors } \; B_p = b_p$$

Les règles de classification fournies par un algorithme d'induction d'arbre de décision appartiennent à cette famille de règles.

Les règles généralisées sont caractées par le fait que leur conséquent est une conjonction de termes attribut-valeur. Elles sont donc de la forme suivante :

$$\text{si } A_1 \; o\text{p } a_1 \text{ET} \ldots \text{ET } A_m \; o\text{p } a_m \;\; a\text{lors } \; B_1 \; o\text{p } b_1 \text{ET} \ldots \text{ET } B_p \; o\text{p } b_p$$

Nous l'avons vu au paragraphe II.2, les algorithmes standard empruntés à l'apprentissage automatique ou conçus spécifiquement pour la fouille de données se basent sur des critères relativement simples et peu sélectifs pour évaluer la qualité des règles : taux d'exemples bien classés, support, confiance. Ils produisent à leur tour un nombre très important de motifs que le volume rend difficiles à interpréter. De plus ils peuvent ignorer des motifs pertinents ; par exemple, le choix du seuil de *support* minimum, dans un algorithme de type APriori, va interdire la sélection de des motifs locaux qui ont éventuellement une bonne confiance et un caractère intéressant. Et par ailleurs, si l'on veut corriger ce défaut en baissant la valeur du seuil, alors le nombre de règles produites risque de rendre le résultat sans intérêt. Cet exemple illustre la difficulté du problème de sélection des meilleurs motifs selon des critères qui ne sont pas indépendants. Le choix du critère ou de l'indice de qualité que l'on veut appliquer à une règle représente également une difficulté. Ces indices sont assez nombreux et ont une incidence directe sur les propriétés de l'espace des règles et donc sur la difficulté même du processus d'extraction. Dans les sections qui suivent, nous présentons un échantillon représentatif des indices proposés pour évaluer les règles selon des critères d'intérêt divers et les caractéristiques observées sur l'espace des règles ainsi définis.

II.3.2. Différents points de vue sur l'*intérêt* des règles

Nous proposons une synthèse des travaux réalisés sur la recherche de règles intéressantes, et, surtout, sur l'étude des indices associés. L'intérêt subjectif, est plutôt défini en termes de nouveauté et d'effet de surprise par rapport à une connaissance *a priori* des experts du domaine. Bien que cet aspect ne doive pas être négligé, nous nous intéressons ici aux critères d'intérêt objectif portant soit sur la forme des règles, soit sur des indices statistiques.

Les travaux sur la recherche de règles d'association se concentrent plus sur la recherche des ensembles d'items fréquents que sur la sélection des règles. Nous devons tout de même évoquer un certain nombre de propositions [BAYA99, NG98, ZAKI00] visant soit à réduire la redondance dans l'ensemble des règles extraites soit à filtrer des règles à partir de critères syntaxiques.
Parmi les travaux relatifs au caractère intéressant des informations extraites, on peut observer une certaine diversité qui s'explique d'une part par la variété des interprétations données à l'*intérêt* et d'autre part, par les différents points de vue que l'on peut adopter. Dans la suite de ce paragraphe, nous présentons les différents travaux sur la recherche de règles intéressantes obtenues par des indices numériques et sur les synthèses et études comparatives réalisées sur ce sujet.

- **Indices d'intérêt**

Assurer l'absence de redondances parmi les règles produites ne suffit pas à donner des règles nécessairement intéressantes. Les critères simples pour la sélection de règle comme la *précision* globale d'un modèle à base de règles ou le *support* et la *confiance* pour des règles prises individuellement sont connues pour être insuffisants si l'on veut extraire des informations utiles et intéressantes.
Pour la qualité globale d'un classifieur, et particulièrement pour les applications médicales ou la recherche d'informations, la *sensibilité (se)* et la *spécificité (sp)* fournissent plus d'information sur la qualité du modèle en donnant des taux de classement sur les exemples positifs (prédits dans la classe) et négatifs (non prédits dans la classe). Rappelons la terminologie employée dans ce contexte : on désigne par le terme de *vrai positif*, un exemple de l'ensemble de test prédit dans une classe par le classifieur et appartenant à cette classe. Un faux positif désigne un exemple de l'ensemble de test prédit dans une classe et n'appartenant pas à cette classe. La *sensibilité* peut alors être considérée comme le taux de vrais positifs par rapport à l'ensemble des exemples de la classe alors que la *spécificité* peut être vue comme le taux de vrais négatifs par rapport l'ensemble des exemples hors de la classe. Ces indices sont particulièrement utiles lorsque la distribution des classes est inégale.
Pour les règles d'association, une des premières mesures objectives plus précises que la *confiance*, *RI (Rule Interest)* a été définie par G. Piatetsky-Shapiro [PIAT91]. Citons d'autres indices proposés pour l'évaluation de la qualité des informations extraites : le *lift* [IBM96], la *J-Measure* de Goodman et Smyth [GOOD91], la mesure définie par Sebag et Schoenauer [SEBA88] ou encore la *conviction* [BRIN97a]. Il a été observé [HAND01] que certaines d'entre elles sont relativement équivalentes.

On utilise l'indice de *précision* ou *taux de succès* sur un ensemble de test c'est-à-dire le taux d'exemples bien classés dans l'ensemble de test pour les modèles exprimés sous la forme d'arbres d'induction à partir d'un ensemble d'apprentissage. On peut également définir la *précision* d'une règle $A \rightarrow B$ en termes de probabilité par $p(A \cap B) + p(\overline{A} \cap \overline{B})$. De même, pour une règle, la *sensibilité* correspond à $p(A/B)$ et évalue la couverture de B par A tandis que la *spécificité* est définie par $p(\overline{A}/\overline{B})$ et évalue la couverture de $\overline{A}$ par $\overline{B}$.

Le choix d'un indice de qualité dépend de l'objectif spécifique du processus d'extraction des connaissances. Par exemple, dans la tâche de prédiction pour un test de diagnostique médical, l'objectif principal est de réduire le taux d'erreur qui consiste à prédire un patient dans la classe "*malade*". Dans ce cas, les experts peuvent considérer qu'il est plus essentiel d'avoir les meilleurs résultats dans la classification d'exemples de la classe "*malade*" même si certains exemples de la classe "sain" sont classés dans "malade". Cela amène à optimiser la *sensibilité* mais non la *spécificité*. Dans d'autres domaines, on peut vouloir optimiser la *sensibilité* et la *spécificité* simultanément. Par exemple, Fidelis et al. [FIDE00] suivent une approche génétique pour optimiser le produit des deux critères.

Pour les règles d'association, maximiser le *support* permet de sélectionner des règles générales, mais la *confiance* a montré ses limites puisqu'elle favorise de nombreuses règles qui sont le plus souvent inutiles. En effet la faiblesse de la *confiance* $p(B/A)$ est de ne pas prendre en compte $p(B)$. De nombreux indices d'intérêt comparent généralement la probabilité *a priori* $p(B)$ et la probabilité *a posteriori* $p(B/A)$; par exemple, le *lift* [IBM96] défini par $p(B/A) / p(B)$ et l'indice *RI (Rule Interest)* [PIAT91] défini par $|A| \times (p(B/A) - p(B))$ mesurent la différence avec la situation d'indépendance entre A et B. Comme le fait remarquer l'auteur dans [BRIN 97], ces deux indices, du fait qu'ils sont symétriques, mesurent la co-occurrence de A et B.

En revanche, la *conviction (conv)* [BRIN 97a] définie par $p(A) \times p(\overline{B}) / p(A \cap \overline{B})$ n'est pas symétrique et mesure l'implication logique $A \rightarrow B$ qui signifie $\overline{A} \vee B$. M. Sebag et M. Schoenauer [SEBA88] ont défini l'indice *Seb* par l'expression $p(A \cap B) / p(A \cap \overline{B})$ qui mesure la proportion entre les exemples positifs et négatifs. En tant qu'indice, la *J-Measure* de Goodman et Smyth [GOOD91] définie par $p(A \cap B) \times T_1 + p(A \cap \overline{B}) \times T_2$, où T_1 et T_2 sont respectivement égaux à $log_2(p(A \cap B) / p(A) \times p(B))$ et $log_2(p(A \cap \overline{B}) / p(A) \times p(\overline{B}))$, en combinant deux indices, mesure la quantité d'information exprimée

par la règle. Cet indice, défini comme l'entropie croisée ne donne pas réellement d'indication sur le lien de causalité de A vers B, mais renseigne sur le degré de correspondance entre les distributions de probabilité. Des valeurs de *J-Measure* élevées sont recommandées, mais elles ne caractérisent pas nécessairement des liens de causalité forts.

Citons également le critère du χ^2 qui est le coefficient statistique du test d'indépendance, le coefficient de corrélation linéaire de Pearson, la mesure de Pearl similaire à RI, la mesure de Loevinger [LOEV47], l'intensité d'implication [GRAS01] et la crédibilité définie par Hamilton [HAMI97]. Parmi tous ces indices, on peut bien sûr observer des propriétés communes ; par exemple deux indices, comme le *lift* et la *conviction*, ont le même sens de variation lorsque l'on fixe les marges $p(A)$ et $p(B)$. Cependant, dan le cas général, les corrélations sont plus difficiles, sinon impossibles à déterminer.

- **Théories et Choix des mesures**

L'état actuel des travaux sur l'*intéressabilité* des règles et motifs découverts reflète une assez grande diversité. Nous avons donné ci-dessus un aperçu des indices proposés et nous citons dans ce paragraphe des travaux plus théoriques ou des travaux de synthèse sur ce sujet. Hilderman et Hamilton [HILD99a, HILD99b, HILD01] proposent un ensemble de principes, axiomes définissant une mesure d'intérêt. Un large échantillon de mesures est testé relativement à la théorie énoncée et montre qu'aucune d'entre elles ne vérifie l'ensemble des propriétés attendues. Cette étude est menée pour des agrégations et ne s'adapte pas directement aux cas des modèles extraits sous forme de règles.

Guillaume [GUIL01] mène une étude intéressante sur le comportement de différents indices vis à vis des cas d'indépendance, incompatibilité, attraction ou répulsion entre antécédent et conséquent de la règle et propose un indice d'intensité d'implication ordinale, extension de l'intensité d'implication pour des règles portant sur des variables numériques ou ordinales.

Lenca et al. [LENC02] posent le problème crucial du choix de l'indice adéquat. Ils proposent de recourir à des méthodes d'aide à la décision pour choisir un ensemble de mesures en fonction de plusieurs critères de choix comme la facilité à placer des seuils ou la non-symétrie de l'indice.

Poursuivant les mêmes objectifs, Lallich et Teytaud [LALL02] étudient les corrélations entre indices à marges fixées et proposent des critères pour aider dans le choix de ces indices. Ils suggèrent d'adapter des outils de la théorie de l'apprentissage statistique pour évaluer un indice.

- **Etudes de l'espace des règles**

Parmi ces travaux sur la recherche de motifs intéressants, très peu sont consacrés aux propriétés de l'espace des solutions (règles). Par exemple, nous pouvons mentionner les travaux de Flockart et Radcliffe [FLOC95] sur une étude comparative menée sur les fonctions d'évaluation implémentées dans le système GA-Miner. Citons également Bayardo et Agrawal [BAYA99] qui proposent de simplifier la recherche de règles d'association intéressantes en se basant sur les propriétés observées sur un espace réduit de règles d'association.

Bayardo et Agrawal étudient l'espace des règles d'association muni d'un ordre partiel défini en fonction du *support* et de la *confiance*. L'étude est implicitement basée sur l'antagonisme qui se vérifie fréquemment entre ces deux mesures, mais qui n'est cependant pas établi en général. Dans cette approche, les auteurs se limitent à certains critères "compatibles" avec le *support* et la *confiance*. Cette étude est limitée aux mesures pour lesquelles la propriété de conservation de l'ordre est observée. Bien que les auteurs utilisent le concept de frontière de Pareto, ils n'y font pas explicitement référence. De plus leur approche est plutôt adaptée au cas où l'on cherche à caractériser la population d'intérêt qui correspond au cas où le conséquent de la règle est fixé.

L'approche de Flockart et Radcliffe [FLOC95] est différente. Plusieurs fonctions d'évaluation des règles sont implémentées dans GA-Miner : *rule interest*, *J-Measure*, *gain d'information*, χ^2 et *coefficient de corrélation*. Une étude expérimentale a été menée sur les distributions de ces fonctions sur l'espace des règles. Les résultats font état de corrélation forte entre ces mesures et de propriétés de symétrie ; les phénomènes particuliers sont observés graphiquement. Cependant le contexte de l'expérience n'est pas décrit ; en particulier, il n'est pas précisé si ces résultats sont obtenus sur tous les jeux de données testés.

II.3.3. Etude de l'espace de recherche des règles : notre approche

L'étude que nous avons menée sur l'espace des règles suit également une approche expérimentale, mais elle ne se limite pas à la recherche de similarité entre mesures et elle considère différents jeux de données. Cela nous a d'ailleurs permis de mettre en évidence le fait que les phénomènes observables peuvent dépendre du jeu de données.

Les propriétés considérées sont utiles non seulement pour compléter l'étude des indices d'intérêt, mais également, pour caractériser et déterminer les algorithmes de recherche les mieux adaptés.

Contrairement aux approches décrites dans la section II.3.2 qui limitent l'espace des règles en fixant des marges, l'expérimentation sur l'espace met en évidence des corrélations impossibles à déterminer de manière analytique.

Nous étudions l'ensemble des règles de dépendance évaluées selon un indice de qualité comme un espace de recherche. Nous observons

- la densité d'états de l'indice c'est à dire la distribution des valeurs de l'indice,
- la répartition des règles selon leur longueur et les valeurs de l'indice,
- les variations des valeurs de l'indice en fonction de la distance à une règle donnée,
- le comportement relatif des indices le plus souvent pris deux à deux.

Dans cette étude, l'objectif est d'étudier certaines caractéristiques de l'espace des règles dans la perspective d'un processus d'optimisation ; il ne s'agit pas de réaliser une étude exhaustive des différents indices proposés dans la littérature. C'est pourquoi nous avons fait le choix des indices de référence suivants : *précision*, *support*, *confiance*, *sensibilité*, *spécificité*, *conviction*, *sebag*, r*ule interest* et *lift*. Un des arguments qui nous paraît essentiel dans le choix de ces indices réside dans le fait que l'utilisateur puisse facilement appréhender leur signification. Ce critère de choix est d'ailleurs mis en exergue dans [LALL02] qui le désigne comme "le sens concret de la mesure". Un deuxième critère de choix pour la majorité de ces mesures est leur caractère dissymétrique : il est important de pouvoir établir une distinction entre les règles $A \to B$ et $B \to A$.

Pratiquement, les jeux de données utilisés sont en grande partie ceux de l'UCI[2] comme *Vote* et *Sonar*. *Vote* décrit le détail des votes de chacun des représentants au congrès des Etats Unis sur des sujets clés ; il contient 435 lignes structurées en 17 attributs booléens comme *physician-fee-freeze, el-salvador-aid, religious-groups-in-schools, aid-to-nicaraguan-contras, mx-missil, superfund-right-to-sue* ou *duty-free-exports.* Ces données sont couramment utilisées pour le classement. L'attribut de classe peut prendre la valeur *democrat* ou *républican*.

Sonar contient la description de 208 signaux de sonar issus de cylindres métalliques (M) ou de pièces de ciment grossièrement cylindriques (R) selon 60 attributs réels et un attribut de classe binaire. Les attributs réels représentent des niveaux d'énergie pour des bandes de fréquences données. Ce jeu est utilisé pour le classement des signaux dans les classes M et R.

Nous avons également utilisé le jeu de données *Churn* qui contient la description de 14945 clients d'une entreprise de téléphonie mobile selon 19 attributs dont 17 attributs de type réel, un attribut de

[2] Université de Californie, Irvine

classe binaire et un attribut catégoriel. Ce jeu a été utilisé pour étudier le phénomène d'attrition, c'est-à-dire pour détecter les clients susceptibles de résilier leur contrat.

Etudier la corrélation entre les valeurs d'indices et une autre mesure est une approche pour caractériser un espace de recherche. Par exemple, B. Manderick et al. [MAND91] étudient le coefficient de corrélation des opérateurs génétiques : ils calculent la corrélation entre le score (fitness) de certains individus et le score de leurs descendants. J. Grefenstette [GREF95] utilise la distribution de score des opérateurs pour prédire le comportement d'un AG. H. Rosé et al. [ROSE96] développent l'approche de la *densité d'état* en considérant le nombre de configurations possédant la même valeur d'indice. T. Jones [JONE95] propose d'utiliser le coefficient de corrélation entre le score (*fitness*) et la distance à l'optimum le plus proche (CFD) comme critère de difficulté d'un processus de recherche sur un espace de recherche donné. Dans notre étude de l'espace des règles, nous reprenons certaines de ces idées comme l'étude de la densité d'états et le CFD. Cependant, la taille des espaces de règles que nous considérons ne permet pas, en général, une étude exhaustive ; il a donc été nécessaire de rechercher une représentation de l'espace par un échantillon. Nous présentons ci-après la méthode d'échantillonnage de Metropolis, puis des résultats les plus remarquables de l'étude des espaces de règles sur les jeux de données choisis.

- **Echantillonnage de l'espace de règles : Méthode aléatoire et Algorithme de Métropolis**

Différentes méthodes peuvent être mises en œuvre pour obtenir un échantillon représentatif de l'espace de recherche. Les travaux sur les densités d'états d'une fonction dans un espace de configurations font par exemple référence aux méthodes analytiques et expérimentales.

Nous avons dans un premier temps utilisé la méthode la plus simple de tirage aléatoire uniforme dans laquelle la présence d'une règle dans l'échantillon représentatif est aléatoire et indépendant des autres. Cette méthode a permis de mettre en évidence la rareté des règles pour des valeurs d'indice élevées. Aussi, pour mieux approcher les "frontières" de l'espace, nous avons utilisé la méthode de Metropolis [ROSE96] qui est basée sur une analogie avec la thermo-dynamique et qui permet d'échantillonner les points de faible fréquence. Cette méthode repose sur une analogie entre les particules d'énergie en thermodynamique et les valeurs de la fonction. Elle se fonde sur la loi thermodynamique qui relie la probabilite d'apparition P(F) de particules d'énergie F à leur nombre N(F) et à la température T par la formule $P(F) \approx N(F)e^{-F/T}$.

Cette méthode procède de la manière suivante :

- à partit d'une configuration aléatoire s_1 de l'espace de recherche, la configuration suivante s_2 est choisie au hasard , par exemple[3]
- si la configuration s_2 est meilleure que s_1 elle est acceptée comme nouvel élément de l'échantillon
- sinon, elle est acceptée avec une probabilité $e^{-\Delta s/T}$ où $\Delta s = f(s_1) - f(s_2)$ et T est la température.

Les paramètres de cette méthode sont la température et la taille de l'échantillon désiré. Les valeurs optimales de ces paramètres ne sont pas connues a priori et doivent être déterminées de façon expérimentale. Concernant la température, on peut commencer par déterminer un intervalle hors duquel le paramètre a peu d'influence sur l'échantillon. A l'intérieur de cet intervalle on opère par valeurs croissantes de la température : une valeur faible aura tendance à figer la recherche vers des solutions de coût croissant, alors qu'une température élevée permettra une plus large exploration. La taille de l'échantillon est également importante ; on procède en général par valeurs croissantes jusqu'à trouver un borne au delà de laquelle la taille n'ait plus d'incidence sur la qualité de l'échantillon.

- **Densité d'états de l'espace des règles**

Nous avons montré dans la section II.3.2 que les études analytiques révélaient des corrélations entre indices ; ces conclusions sont obtenues en considérant un espace de règles réduit dans lequel on fixe certaines données, par exemple, la cardinalité du conséquent de la règle. Par contre, l'étude expérimentale que nous avons menée a consisté à étudier la distribution des valeurs d'un indice sans restriction sur l'espace des règles.

A titre d'exemple, nous présentons les résultats obtenus pour les indices de *conviction*. La Figure II-1 représente respectivement la densité d'états de cet indice sur un échantillon Metropolis de *Churn*. On observe la faible fréquence des valeurs optimales de *conviction* expliquant la difficulté à découvrir une règle de *conviction* optimale. On observe également que la majorité des règles ont une valeur de *conviction* faible à moyenne et que la valeur moyenne de conviction est proche de 1. On peut en déduire que la valeur de *conviction* d'une règle tirée au hasard serait également proche de 1, ce qui traduit une situation d'indépendance entre l'antécédent et le fait d'être un contre-exemple de la règle

[3] cette configuration pourrait également être choisie dans un voisinnage

(exemple qui ne vérifie pas le conséquent). La longueur de la "queue" de la distribution montre que le problème peut être difficile : plus on se rapproche de la valeur d'indice maximale, plus les règles deviennent rares. De manière plus générale, ce critère peut être utilisé pour caractériser la difficulté ou comparer l difficulté de deux espaces de recherche.

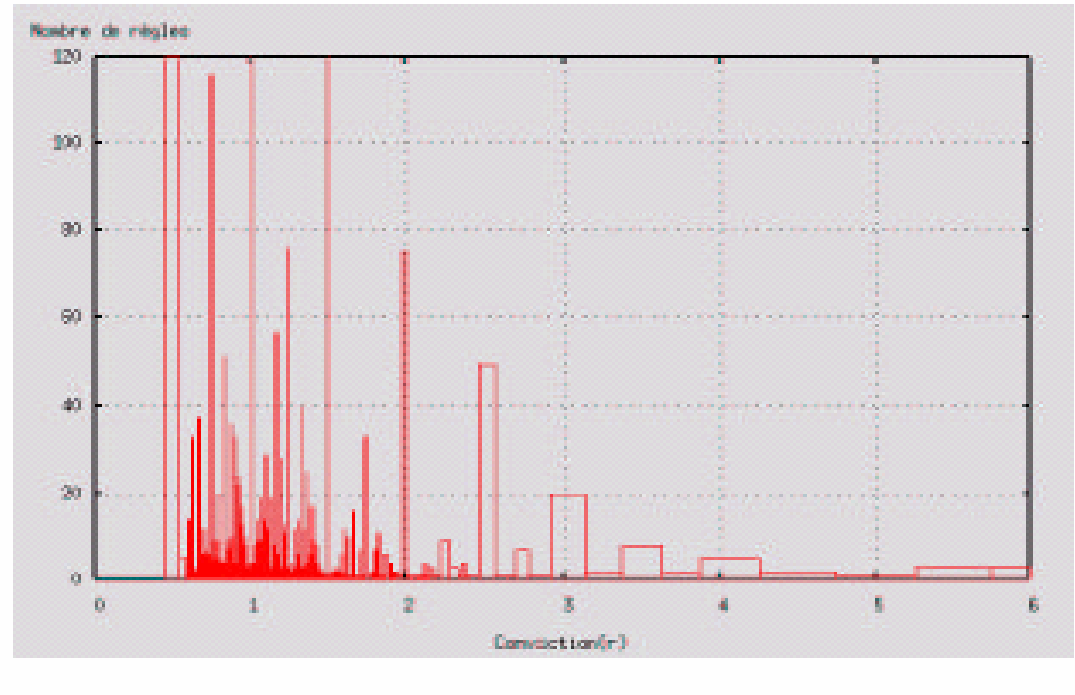
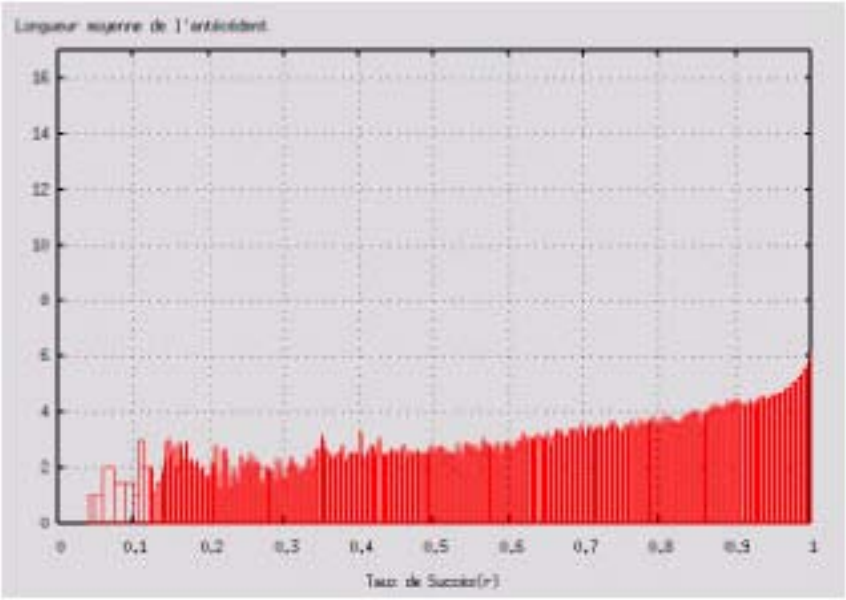


Figure II-1 Distribution de la *conviction* des règles simples sur *Churn*

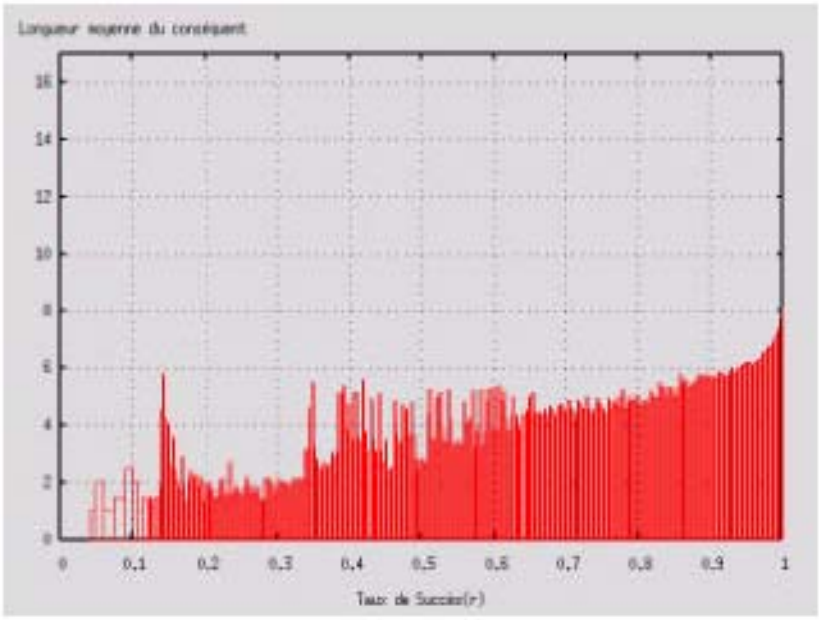
- **Valeurs d'indices versus Longueur des règles**

Nous avons évoqué à la section II.3.2, le critère syntaxique de longueur pour la sélection des règles intéressantes. En effet, il est énoncé comme un principe d'intelligibilité que la longueur des règles doit rester faible. Dans les algorithmes d'induction d'arbres de décision, la phase d'élagage permet effectivement de réduire la longueur des chemins jusqu'aux feuilles et donc la longueur des règles déduites. Nous avons cité en section II.3.2, les travaux sur les *évènements rares* qui tendent à démontrer l'intérêt de ces motifs. Or les règles dont l'antécédent est composé de nombreux termes représentent généralement peu d'exemples du jeu de données et constituent ainsi des *évènements rares.*

Aussi l'observation de ces règles *locales* qui couvrent peu d'exemples, mais qui peuvent révéler des informations pertinentes vient s'opposer au principe selon lequel les règles intéressantes doivent être recherchées parmi les règles courtes.



(a)



(b)

Figure II-2 Longueur des antécédents (a) et conséquents (b) des règles généralisées selon la *précision* sur *Vote*

Les Figure II-2 et Figure II-3 représentent la distribution des longueurs de règle respectivement pour la *précision* et la *spécificité*. On peut ainsi observer la présence de règles longues et cependant intéressantes selon ces critères. Cette information est utile si l'on recherche des motifs très locaux ou évènements rares pertinents, couvrant peu d'exemples, qui ne sont pas découverts par une méthode d'induction d'arbre de décision par exemple. Notons que des graphes représentant les longueurs de règles sur d'autres indices et jeux de données n'ont pas permis d'observer le même phénomène. Ces résultats suggèrent qu'il n'est pas souhaitable d'orienter la recherche vers des règles soit courtes, soit longues car on ne connaît pas, a priori, la longueur des règles intéressantes. Il serait utile de disposer d'une méthode de recherche qui ne biaise pas le processus de recherche vers des règles de longueur donnée.

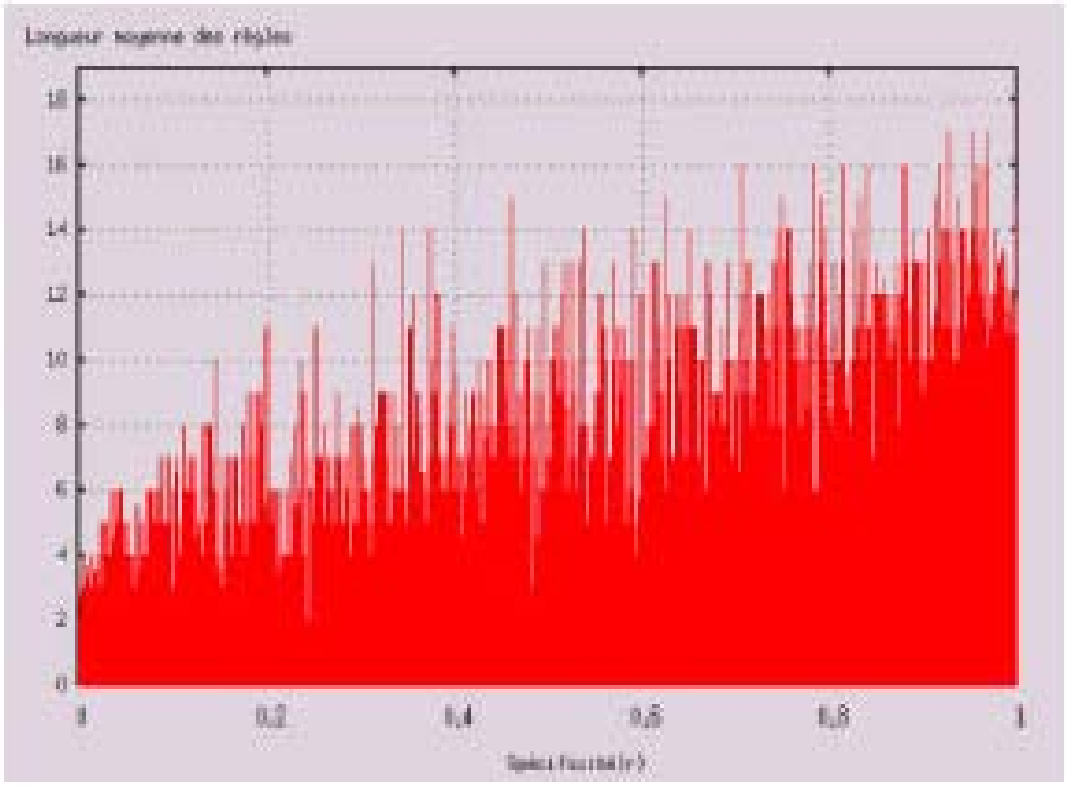


Figure II-3 Longueur des règles généralisées selon la *spécificité* sur *Churn*

- **Variations des valeurs d'indice**

Bien que la méthode que nous proposons ici soit inspirée par la mesure de corrélation entre la fitness et la distance au plus proche optimum (CFD), notre objectif est différent. Nous étudions la présence d'optima locaux de façon à mettre en évidence l'oubli de "bonnes" règles par un algorithme de recherche locale.

La Figure II-4 montre les variations de valeurs de *précision* sur des règles générées à partir de *Churn*. La représentation graphique est obtenue de la manière suivante :

- une règle r_0 est choisie parmi les meilleures règles produites par un algorithme d'induction[4] d'arbre de décision qui effectue une recherche locale
- une autre règle r de l'échantillon est représentée en fonction de sa distance à r_0

La distance entre deux règles r_1 et r_2 est égale au nombre d'attributs apparaissant soit uniquement dans r_1, soit uniquement dans r_2 (on exclut donc les attributs communs aux deux règles). On donne ainsi un sens précis à la notion de *localité*. Cette définition est cohérente puisque le voisinage induit par cette distance, pour une règle est celui auquel l'on fait implicitement référence lorsque l'on parle de recherche locale. En abscisse, figure la distance à la règle r_0 et en ordonnée, la valeur d'indice. La règle r_0 est parmi les règles de meilleure *précision* découvertes par l'algorithme d'induction d'arbre ; elle est représentée sur l'axe des ordonnées par sa valeur de *précision*. On peut observer l'enveloppe supérieure du nuage de points constituée des maxima. Elle fait apparaître la règle r_0 comme un maximum local, ce qui confirme que C4.5 ne découvre pas le maximum global. Sur cette même figure, les points bleus figurent la zone de l'espace dans lequel l'algorithme recherche la meilleure règle en construisant incrémentalement la solution, attribut par attribut.

- **Corrélations et antagonismes**

Certaines mesures sont connues pour être fortement corrélées ; c'est la cas, en particulier si l'on restreint l'espace des règles $A \rightarrow B$ en fixant la probabilité du conséquent. Par exemple, le *support* et l'indice *Rule Interest* trient les règles selon le même ordre si l'on fixe p(B).

[4] l'implémentation J48 de l'algorithme C4.5 fourni dans le système WEKA

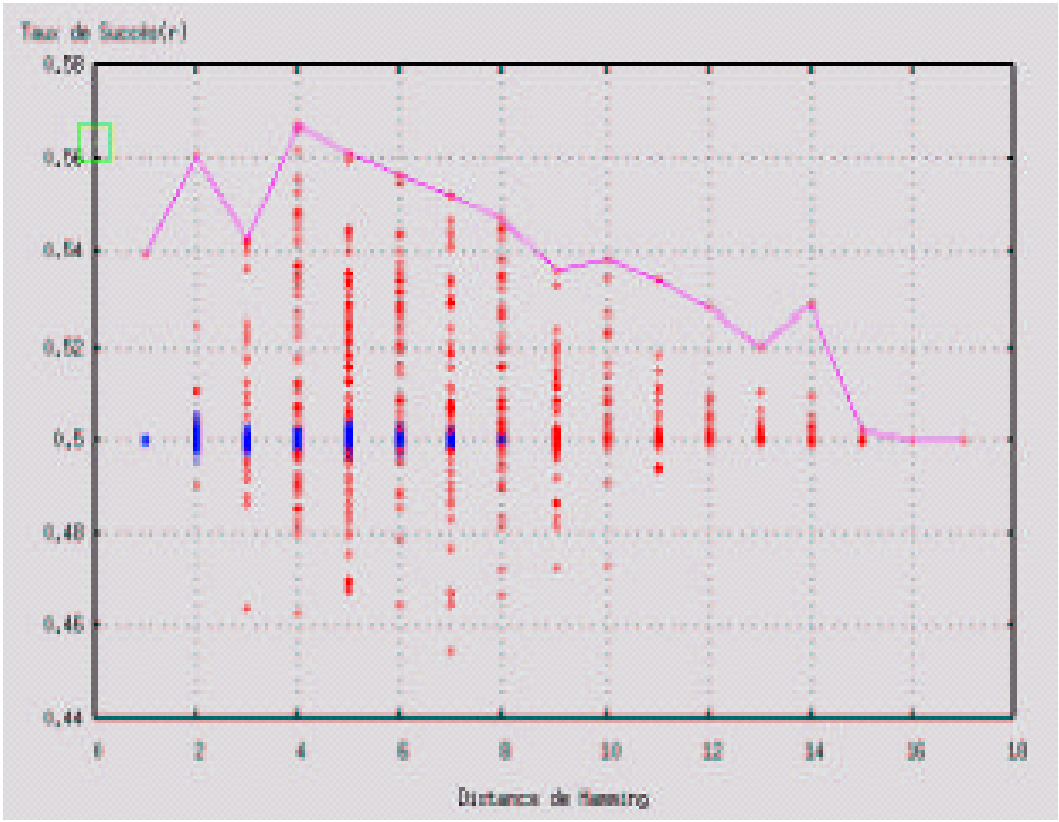
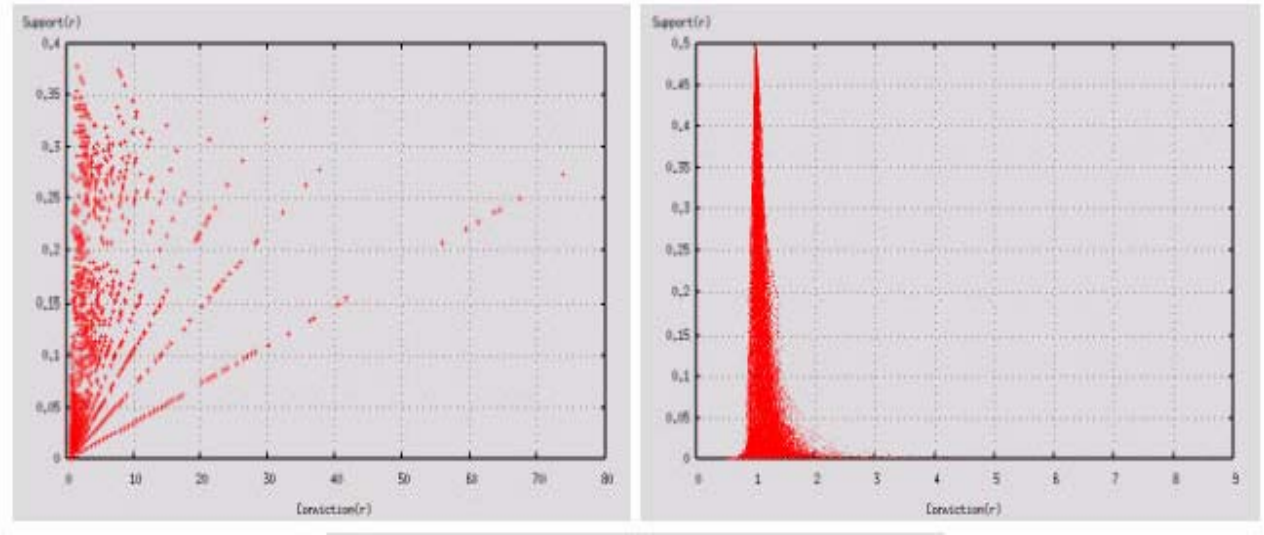


Figure II-4 Variations des valeurs de *précision* des règles simples sur *Churn*

[LALL02] montre qu'il en est de même pour *lift*, *conviction* et *Seb*. La *sensibilité* et la *spécificité* sont des indices complémentaires de couverture ; les courbes ROC exploitent d'ailleurs cette propriété dans le but d'évaluer les performances d'un classifieur pour différents seuils d'appartenance à la classe. Notons que notre approche est différente puisque nous nous intéressons à l'évaluation individuelle de règles et non pas à un ensemble de règles évaluées collectivement.

Une analyse formelle de ces mesures permet de tirer des conclusions partielles, mais ne permet pas de mettre en évidence des propriétés générales liant ces différentes mesures. En revanche, l'approche expérimentale que nous suivons, met en évidence graphiquement, les comportements relatifs de ces mesures prises deux à deux sur différents jeux de données. Nous avons pu de cette manière, observer des corrélations ou des antagonismes entre mesures selon le jeu de données.

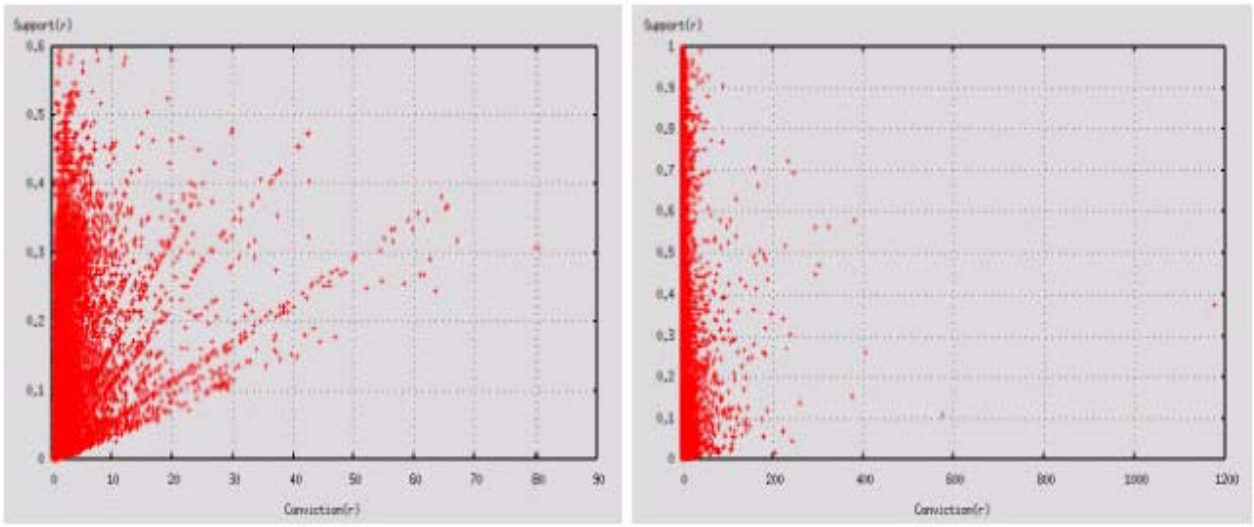
Notons que dans un but de simplicité, nous considérons ici des couples de mesures, mais la démarche pourrait s'adapter à un nombre plus élevé de mesures. Dans l'étude expérimentale menée [FRAN03a], nous avons identifié de nombreux cas où des antagonismes sont apparents. Nous présentons ici deux exemples. La Figure II-5 met en évidence l'antagonisme entre le *support* et la *conviction* apparent sur les jeux de données *Vote* et *Churn*. Cet exemple démontre l'intérêt d'une approche qui identifierait directement les bons compromis parmi les couples (*support*, *conviction*) par rapport aux solutions inspirées de l'algorithme APriori procédant séquentiellement en deux étapes (*support* puis *conviction*).
La Figure II-6 présente le comportement relatif de *rule interest* et *sebag* qui sont également complémentaires sur *Vote* et *Churn*. Nous avons également observé les indices de *sensibilité* et *spécificité* qui mesurent respectivement la précision et la non-couverture des exemples négatifs et qui, considérés simultanément permettent d'identifier des règles $A \rightarrow B$ pour lesquelles l'antécedent A caractérisent entièrement le conséquent.



(a) **(b)**

Figure II-5 Règles simples évaluées selon *support* et *conviction*

sur *Vote* (a) et *Churn* (b)



(a) **(b)**

Figure II-6 Règles simples évaluées selon *sebag* et *RI* sur *Vote* (a) et *Churn* (b)

II.4. Algorithmes de recherche

On peut rappeler le théorème dit du *no-free-lunch* (NFL) de Wolpert et Mac Ready [WOLP96] selon lequel il ne peut pas exister un algorithme qui apporte une solution à tout problème d'optimisation et qui soit supérieur à tous les autres. Pour un type d'algorithme, on peut simplement identifier des classes de problèmes spécifiques pour lesquelles il est meilleur, et également des problèmes pour lesquels il n'est pas performant.

Les caractéristiques de l'espace des règles qui nous intéressent et les limites des algorithmes couramment mis en œuvre orientent notre choix vers les algorithmes évolutionnaires (AE). En effet, espace de recherche très large, présence d'optima locaux, fonction d'évaluation complexe et variable, mesurs multi-critères caractérisent des problèmes sur lesquelles les AE sont connus pour être efficaces. Bien que marginaux dans le cadre des recherches en fouille de données, ils représentent, si nous adoptons un point de vue "optimisation", une alternative prometteuse.

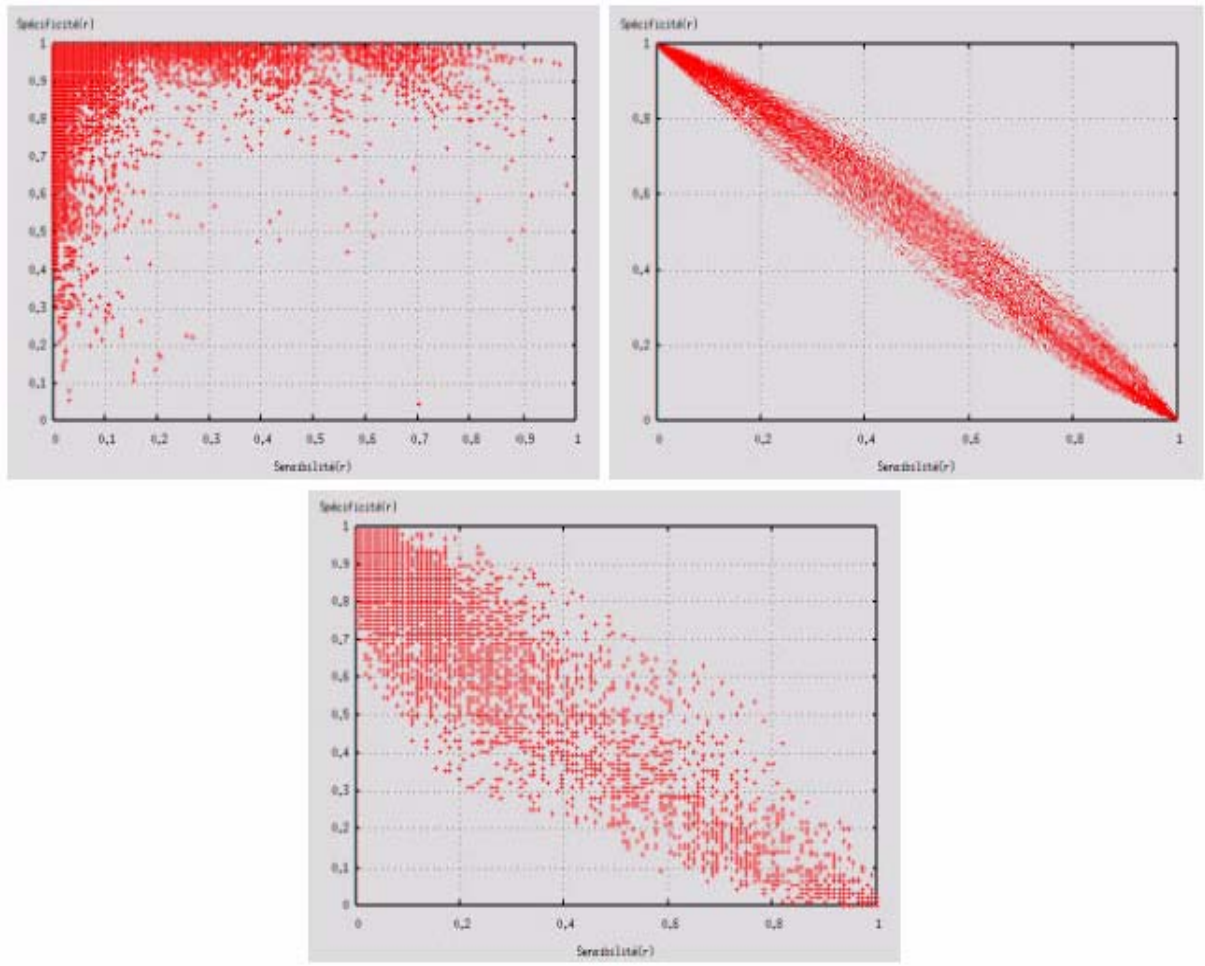


Figure II-7 Règles généralisée évaluées selon *se* et *sp* sur *Vote*, *Churn* et *Sonar*

Les algorithmes évolutionnaires sont des techniques de recherche stochastiques inspirées de l'évolution des espèces naturelles. On peut dire, de manière générale qu'ils se réfèrent à la théorie selon laquelle les organismes vivants s'adaptent à leur environnement et deviennent meilleurs (mieux adaptés). Cette adaptation se manifeste par le fait qu'un organisme devienne capable de transmettre ces gènes à ses descendants, ce que Darwin a appelé "the survival of the fittest". La sélection naturelle montre son aptitude à résoudre un problème qui consiste à produire des organismes de mieux en mieux adaptés. Les AE tentent de simuler, de façon certes sommaire, les systèmes d'évolution naturelle sur une machine et ainsi d'utiliser leur capacité de résolution sur leurs propres problèmes de recherche et d'optimisation.

Contrairement à la majorité des techniques de recherche qui démarrent en choisissant un point de l'espace de recherche et poursuivent en opérant des déplacements locaux à partir de ce point, un AE

maintient une population de points de l'espace. L'étape d'initialisation ne commence pas par un seul point, mais par une population qui peut être choisie aléatoirement. Une autre différence fondamentale avec les techniques courantes réside dans le fait que l'on puisse générer de nouvelles solutions, non seulement par de petites variations, mais également par recombinaison de deux solutions. Les AEs utilisent donc un opérateur binaire, dit *de croisement*, qui permet de combiner deux parents pour produire des descendants. Une autre façon de produire un descendant est d'appliquer un opérateur unaire de modification à la nouvelle solution. Cet opérateur est appelé *opérateur de mutation* et s'inspire des changements génétiques aléatoires qui se produisent dans l'évolution naturelle.

Il y a deux manières d'appliquer la *sélection* sur les solutions : on peut sélectionner les parents avant croisement ou bien, lors du remplacement par un descendant, on peut sélectionner la solution à remplacer.

Il existe différents types d'algorithmes évolutionnaires ; les plus connus sont les *algorithmes génétiques* (AG) introduits par John Holland [HOLL75] ; à leur origine, ils étaient plutôt utilisés pour des problèmes d'optimisation combinatoire où les solutions sont représentées par des chaînes binaires et la sélection est appliquée sur les parents.

Les *stratégies d'évolution* développées par Rechenberg et Schwefel sont plutôt utilisées sur des problèmes d'optimisation continue et considèrent l'opérateur de mutation comme opérateur principal. Elles tendent à appliquer la sélection sur le descendant qui va survivre.

La *programmation génétique* promue par J. Kosa [KOS92] est essentiellement basée sur l'application des algorithmes génétiques à des structures arborescentes pour soumettre des programmes à une évolution génétique. Les nombreuses expériences auxquels s'est prêté Koza montrent qu'il est possible de faire évoluer, des programmes Lisp jusqu'à ce qu'ils résolvent le problème pour lequel on les sélectionne.

Pourquoi s'intéresse-t-on aux algorithmes qui simulent l'évolution? Parce qu'ils sont très flexibles, s'adaptant à de nombreux problèmes en choisissant une représentation des solutions et une fonction d'adaptation (*fitness*) adéquates. Citons aussi D. Fogel [FOGE00] : "parce que les alternatives manquent en matière d'optimisation ; on ne peut pas aisément utiliser les méthodes d'optimisation classiques en présence de maxima globaux entourés de nombreux optima locaux".

Un AG manipule des solutions ou *individus* constitués d'une séquence de gènes (*chromosome*), alors que les organismes biologiques peuvent en posséder plusieurs. L'approche informatique idéalise traditionnellement le chromosome en le réduisant à un vecteur de gènes. Dans le cadre d'un AG, l'individu est réduit à son génome, ensemble de caractéristiques élémentaires. Un *gène* est une

caractéristique d'une solution au problème traité. Le *locus* est la position du gène sur le chromosome. Une *population* est un groupe d'individus.

Dans le cadre de cette étude, nous utilisons uniquement des algorithmes génétiques. En tant que méthode de recherche, l'approche génétique consiste à manipuler un ensemble de solutions potentielles auxquelles on attribue des valeurs (scores). En considérant une solution comme un individu et en assimilant les caractéristiques d'une solution aux composants d'un génome, l'AG fait évoluer une population de solutions en y propageant les caractéristiques des individus les plus viables qui, suivant la métaphore évolutionniste, sont les plus adaptés. Les individus ainsi soumis à une sélection et une évolution vont s'adapter et trouver une niche écologique qui correspond à un optimum dans l'espace de recherche. Les maxima locaux peuvent en outre être surmontés car les individus sont, tout comme dans la nature, soumis à des mutations aléatoires engendrant de nouvelles caractéristiques. Un algorithme génétique soumet une population d'individus à une sélection puis aux opérateurs génétiques suivants :

- la *reproduction* qui consiste à dupliquer un individu dans la population, donc à copier ses gènes,
- le *croisement* qui recombine deux individus parents en produisant deux nouveaux individus, descendants qui héritent des caractéristiques parentales,
- la *mutation* qui est une altération aléatoire des gènes d'un individu.

Le cycle d'activation d'un algorithme génétique peut se décomposer en quatre étapes :

1. évaluation des individus d'après leurs caractéristiques
2. reproduction des individus par sélection aléatoire pondérée en fonction de la valeur obtenue en 1
3. croisement entre individus de la nouvelle génération
4. application de mutations aléatoires

La *mutation* est un processus qui met en jeu un seul chromosome et provoque une modification de certains de ses gènes. Le gène modifié peut provoquer un accroissement ou un affaiblissement de la valeur de la solution que représente l'individu. Ce mécanisme est similaire à celui que l'on observe dans la nature si l'on omet l'existence de gènes récessifs ou dominants qui peuvent en limiter l'effet. Une mutation peut procurer à l'individu, un avantage en terme de sélection et son caractère aléatoire peut permettre d'échapper à un maximum local. L'opérateur de mutation est appliqué avec une probabilité faible, par exemple de l'ordre de 1/N où N représente la taille des chromosomes.

L'opérateur de *croisement* est la transposition informatique du mécanisme qui permet, dans la nature, la production de chromosomes qui héritent partiellement des caractéristiques de leurs parents. Lors d'un croisement, l'information génétique est globalement préservée, les gènes étant transférés d'un individu à l'autre. C'est le croisement qui permet l'exploitation, par recombinaison, des "briques" de bases sélectionnées au cours des générations. L'opérateur de croisement est appliqué avec une probabilité de l'ordre de 0.8.

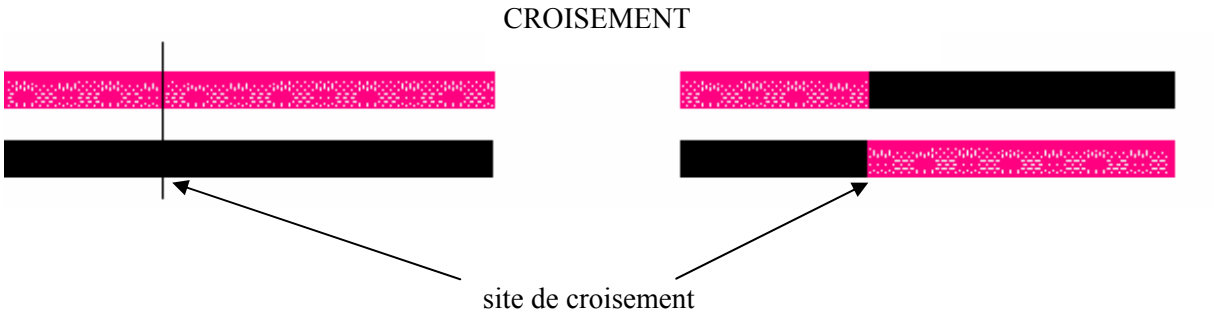


Figure II-8 Croisement de deux individus : parents et descendants

Pour ce qui concerne le *codage des chromosomes*, le large domaine d'application de l'approche génétique résulte du fait que toute information, donc toute solution au problème posé, peut se représenter sous la forme d'une séquence de symboles. Le concepteur doit choisir le codage des solutions au problème qu'il désire résoudre. Il est ainsi possible d'appliquer un AG à des fonctions de plusieurs paramètres. Dans ce cas, il suffit de gérer une population dont les individus sont la concaténation de caractéristiques simples. Chaque solution du problème peut être représentée sous la forme d'un vecteur binaire dans lequel chaque paramètre est codé séparément. Le codage des solutions demande un grand soin. En effet, la valeur d'une solution repose en général sur la présence de certaines caractéristiques. Etant donné que la tâche de l'AG est de propager les bonnes associations de caractéristiques au sein de la population, il est nécessaire d'avoir une idée de la survie de ces associations sous l'effet des différents opérateurs génétiques.

Un AG est contrôlé par plusieurs *paramètres*:

- le codage des chromosomes
- la fonction d'évaluation qui détermine la probabilité de sélection et de reproduction d'un individu,
- la taille de la population
- la probabilité de mutation que subit chaque locus lors de la reproduction,

- la probabilité de croisement qui détermine la fréquence à laquelle les hybridations entre individus vont avoir lieu.

Alors que la mutation permet l'apparition de nouveaux gènes par la modification d'un allèle, le croisement diffuse les gènes existants lors du renouvellement de la population. Cette balance entre l'exploration et l'exploitation doit être soigneusement respectée car un taux de mutation trop élevé entraîne la destruction de gènes avant qu'ils n'aient eu la possibilité d'être assemblés par croisement afin de former des structures valables. Si le taux de croisement est excessif, la population s'uniformise trop rapidement et la population converge alors prématurément vers un type d'individu probablement sous-optimal.

La métaphore des *niches écologiques* peut être utilisée pour éviter la convergence vers des optima locaux. Si un problème connaît plusieurs optima, l'expérience montre qu'un AG standard sera attiré par un seul de ces optima. L'optimisation multimodale consiste à localiser plusieurs optima dans l'espace de recherche. Cela nécessite de maintenir des sous-populations aux voisinages respectifs des différents optima. La notion suggérée par J.Holland de partage des ressources limitées au sein d'une niche écologique est l'une des plus efficace pour maintenir de telles sous-populations. Goldberg [GOLD89] a proposé une implémentation de ce concept connue sous le nom de méthode de partage (*sharing*). Une notion de similarité doit être introduite entre individus ; elle est définie en fonction d'une distance au niveau génotypique ou phénotypique. La méthode consiste à attribuer à chaque individu, une valeur d'adaptation partagée égale à son adaptation brute pondérée par une quantité d'autant plus grande qu'il y a d'individus qui lui ressemblent. A adaptation brute égale, un individu isolé aura plus de descendants qu'un individu qui possède de nombreux voisins.

Pour la *découverte de règles* par algorithmes génétiques, différentes techniques ont été proposées ; elles se répartissent en deux catégories qui se distinguent par la manière dont les règles sont représentées dans la population.

- Approche *Pittsburgh* versus *Michigan*

L'approche *Pittsburgh* introduite par K. Dejong, consiste à considérer un individu comme un ensemble de règles. En revanche, l'approche de l'Université du *Michigan* considère chaque règle comme un seul individu. Lors de l'application de l'AG, la base de règles est utilisée en tant que population et les opérateurs génétiques sont appliquées au niveau de la règle. Le choix entre les deux approches dépend du résultat souhaité. Si l'on recherche un ensemble de règles constituant un classifieur qui doit être évalué dans son ensemble, l'approche de *Pittsburgh* est mieux adaptée. Par contre l'approche Michigan

est plus naturelle lorsqu'on s'intéresse à la qualité individuelle des règles considérées de manière indépendante. Nous avons suivi cette dernière approche dans les travaux expérimentaux que nous avons menés et qui sont présentés au paragraphe II.4.2. Dans cette approche, il y a deux possibilités pour découvrir des règles. La première qui est la plus simple, consiste à exécuter l'algorithme génétique pour découvrir une seule règle. Ainsi, pour découvrir un ensemble de règles, il est nécessaire de réaliser plusieurs exécutions. La seconde possibilité consiste à rechercher directement un ensemble de règles. Nous avons choisi cette dernière approche dans les expérimentations qui suivent. La principale difficulté associée à l'approche de Michigan dans ce cas, est de trouver l'ensemble des meilleures règles sans que l'algorithme ne se focalise sur un seule bonne règle. Pour prendre en compte ce problème, nous avons appliqué une technique de "niching".

- Codage d'une règle

Pour une règle, il existe plusieurs manières de coder l'antécédent et le conséquent. En général, un gène correspond à un attribut et représente une condition composée d'un opérateur et d'une valeur. Si le codage binaire est choisi et si l'attribut est binaire ou catégoriel, une chaîne de bits est associée à chacune de ses valeurs. La démarche est la même pour l'opérateur. Lorsque l'attribut est continu, les valeurs de son domaine sont discrétisées de manière à retrouver le cas d'un attribut catégoriel. Cette étape de discrétisation est un pré-traitement effectué par des algorithmes de discrétisation, qui généralement ne prennent pas en compte l'interaction entre attributs. Ceci constitue un problème car l'algorithme génétique peut alors ignorer d'éventuelles interactions entre attributs. Cependant, les AG peuvent aussi utiliser directement au sein des individus un codage réel pour les attributs de ce type. L'avantage du codage binaire est la simplicité des opérateurs génétiques associés ; par contre il tend à augmenter la taille des chromosomes alors que le codage direct facilite la représentation, mais nécessite de définir des opérateurs génétiques spécifiques plus complexes.

En ce qui concerne le conséquent d'une règle, on peut le coder directement dans le génotype. Remarquons que pour une règle de classification dans une classe fixée, le codage du conséquent devient inutile si l'on exécute l'algorithme pour chaque classe.

La section qui suit présente les recherches en fouille de données et approches évolutionnaires.

II.4.1. Approches évolutionnaires en fouille de données

Les travaux de fouille de données qui font appel à des algorithmes génétiques sont peu nombreux. Ils sont consacrés à la découverte de règles, à la recherche de clusters ou à la sélection de caractéristiques. Dans cette section, nous présentons une synthèse de ces travaux.

Dans [BHAT99], S. Bhattacharya propose un AG pour une tâche de classement assez courante. Il s'agit d'identifier les clients les plus susceptibles de répondre à une campagne de publicité par mailing postal. L'objectif est de minimiser les frais de mailing en sélectionnant la plus petite (ou la moins profonde) et la meilleure proportion de clients. L'intérêt de l'AG dans ce cas est de permettre de considérer la profondeur du mailing (quantile) comme un paramètre et d'optimiser le taux de réponse pour différents niveaux de profondeur. Dans [BHAT00], l'auteur reprend cet objectif avec un AG multiobjectifs pour optimiser simultanément, le taux de réponse ainsi que la précision du modèle découvert.

Dans le système GLOWER [DHAR00], un AG est utilisé pour une tâche de prédiction dans le domaine financier. L'objectif est de rechercher des interactions insoupçonnées dans des données volumineuses. Les auteurs se basent sur les indices de *support* et de *confiance* pour évaluer la qualité d'une règle. Ils évoquent la supériorité de l'AG pour la découverte de règles par rapport à un algorithme glouton d'induction d'arbre par exemple.

L. Jourdan et al. [JOUR02] utilise un AG pour extraire des informations en génomique. L'avantage de l'AG, dans ce contexte, réside dans sa capacité à parcourir des espaces de recherche très larges comme c'est souvent le cas dans ce domaine.

Dans le système SIA [VENT93], un algorithme génétique permet de découvrir des règles logiques d'ordre un. La fonction d'évaluation est exprimée comme une somme pondérée de plusieurs indices dont le taux d'exemples positifs, la généralité de la règle et des indices traduisant les préférences de l'utilisateur. L'auteur souligne l'avantage d'un AG par rapport à un algorithme d'induction d'arbre qui effectue une recherche locale.

GA-MINER [RADC94, FLOC95] est un outil général de recherche de modèles permettant différents niveaux de supervision pour l'utilisateur ; on peut, en particulier rechercher des modèles à bases de règles. Plusieurs fonctions d'évaluation sont implémentées comme le coefficient de corrélation de Pearson, le χ^2 ou la J-Measure. Un algorithme génétique a été parallélisé de manière à offrir des performances viables sur des données volumineuses.

A. Freitas et al. [FREI99, FREI02, NODA99, FIDE00] est l'auteur de nombreuses publications sur la mise en œuvre d'algorithmes évolutionnaires en fouille de données. Il s'intéresse en particulier aux différents indices d'intérêt et utilise des fonctions d'évaluation combinant plusieurs indices. Ses travaux recouvrent des aspects variés de l'extraction de règles et des problèmes spécifiques posés en fouille de données. On peut noter à ce sujet la proposition d'un algorithme hybride [CARV00b] permettant de découvrir des règles générales en induisant un arbre d'induction ainsi que des règles locales extraites par un AG.

II.4.2. Approche expérimentale

Dans la plupart des travaux présentés ci-dessus, les objectifs restent liés au contexte d'application et les expériences se limitent à un sujet précis. Seul, A. Freitas multiplie les expériences autour des algorithmes évolutionnaires en fouille de données ; sans toutefois mettre en avant le point de vue optimisation.
Notre approche trouve son originalité dans ce changement de point de vue qui justifie l'intérêt de recourir aux techniques évolutionnaires lorsque l'espace de recherche s'y prête. Nous présentons l'algorithme mis en oeuvre ainsi que les résultats obtenus.

- **Algorithme**

Notre objectif est de concevoir un AG qui recherche le meilleur ensemble de règles de dépendances selon un critère donné. Un individu représente une unique règle. Un mécanisme de niches est mis en place pour assurer la diversité des résultats. Les opérateurs de mutation et de croisement sont adaptés aux deux types de règles : règles simples et règles généralisées. Un individu, qui correspond à une règle, est représenté par une liste, de longueur fixe, de gènes de la forme *a o v p*. Le booléen *p* indique la présence du gène ou son absence ce qui permet à l'individu, bien qu'ayant un génotype de taille fixe, de représenter une règle de taille variable. L'étude de l'espace des tailles de règles présentée au paragraphe précédent, a montré la nécessité d'explorer tout l'espace de ce paramètre. Ce codage permet de faire varier la taille d'une règle par mutation et favorise donc l'exploration de règles de tailles variées. Nous avons utilisé deux représentations différentes selon que la recherche est dirigée vers des règles simples ou des règles généralisées. Dans le premier cas, la partie *conséquent* n'est pas représentée dans l'individu.

L'opérateur de croisement est adapté au type des règles, simples ou généralisées. Pour le croisement des règles simples, on commence par déterminer un site de croisement de façon aléatoire puis l'on échange les matériels génétiques des deux individus. Pour les règles généralisées, le croisement doit garantir que les individus obtenus restent valides. L'opérateur de mutation s'applique de la façon suivante : un locus est sélectionné de façon aléatoire. Si le gène à ce locus est inactif c'est-à-dire que l'attribut *p* est à 0, celui-ci est rendu actif, et un opérateur ainsi qu'une valeur sont générés de façon aléatoire. Si le gène est présent, soit il est rendu absent, soit son opérateur et sa valeur soient réinitialisés. Pour les règles généralisées, on doit également garantir que les individus obtenus soient valides.

La sélection s'effectue par un tournoi entre deux individus. Le tournoi est l'opérateur de sélection le plus populaire. Il consiste à choisir aléatoirement un groupe de 2 individus et à les faire prendre part à un tournoi basé sur leur fitness. Seul, le gagnant est inséré dans la population suivante.

- **Fonction d'adaptation**

La fonction d'adaptation (fitness) peut être réduite à une mesure d'intérêt ou bien être construite comme une combinaison d'indices selon les objectifs de l'utilisateur. L'évaluation d'une règle nécessite un parcours du jeu de données.

Nous avons dans un premier temps conduit des expériences visant à optimiser un seul indice. Mais, pour prendre en compte plusieurs indices, nous avons également considéré des fonctions sous forme de combinaisons de deux ou plusieurs mesures. Nous avons testée une méthode d'optimisation scalaire où la fitness à optimiser pour une règle est une somme pondérée des mesures à considérer simultanément.

L'algorithme génétique utilisé met en œuvre un mécanisme de niches écologiques pour promouvoir la diversité et pour éviter que la population finale ne soit pas dominée par un seul individu. Cependant cette méthode de combinaison présente deux inconvénients :

- d'une part, il est nécessaire d'exécuter l'algorithme pour chaque ensemble de poids ce qui généralement conduit à des temps de calcul prohibitifs.
- d'autre part, le choix des poids est arbitraire et donc introduit un biais dans le processus d'optimisation

Nous avons donc recherché une méthode mieux adaptée à l'optimisation multicritères.

- **Approche multicritères**

Parmi les nombreuses méthodes d'optimisation multicritères [DEB01, COLL02], méthodes scalaires, méthodes interactives, méthodes floues … les metaheuristiques sont souvent utilisées pour résoudre des problèmes d'*optimisation difficile*. En particulier, les algorithmes génétiques sont bien adaptés à l'optimisation multicritères, essentiellement parce qu'ils manipulent un ensemble de solutions potentielles et qu'ils peuvent exploiter la relation de *dominance de Pareto*.

L'optimisation multi-objectif ou l'optimisation de *Pareto* peut être définie mathématiquement de la façon suivante : chaque solution X_i est associée à un vecteur d'évaluation $F = (F_1(X_i), ..., F_N(X_i))$ où N est le nombre d'objectifs (ou de critères). On dit qu'une solution X_1 *domine* une autre solution X_2 si

$$\forall j : F_j(X_1) \geq F_j(X_2) \text{ et } \exists k : F_k(X_1) > F_k(X_2),$$

où F_j et F_k sont respectivement le $j^{ème}$ et le $k^{ème}$ objectif.

Aucune des deux solutions ne domine l'autre si

$$\exists m_1, m_2 : F_{m_1}(X_1) > F_{m_1}(X_2), F_{m_2}(X_2) > F_{m_2}(X_1)$$

La *frontière de Pareto* est définie comme l'ensemble des solutions non dominées. Les fonctions d'objectifs sont ici des mesures de qualité et les variables X_i représentent les règles. L'algorithme NSGA (Non dominated Sorting Genetic Algorithm) [SRIN95] qui est reconnu pour garantir la diversité dans l'ensemble des solutions. Cette section présente cet algorithme qui utilise la dominance au sens de Pareto pour déterminer la fitness d'un individu. A chaque individu est associé un rang qui est calculé à partir du nombre d'individus de la population courante qui le domine. Ce rang est défini de la façon suivante pour un individu x : $rang(x) = 1 + p(x)$ où $p(x)$ est le nombre d'individus qui dominent l'individu i. Ainsi, un individu non-dominé se voit assigné le rang 1. Les individus de rang P appartiennent à la catégorie à laquelle est assignée la valeur de fitness de $1/P$. Pour obtenir une diversité des individus dans cette catégorie (ou niche), la valeur de fitness des individus est calculée par la formule suivante : $fitnessEffective(x) = fitness(x)/m_x$ avec $m_x = \sum_{y=1}^{K} P(d(x,y))$

et $P(d(x,y)) = 1 - (d(x,y)/disInf)$ *si* $d(x,y) < disInf$ *et 0 sinon*

Le terme *fitness* représente ici la valeur assignée à la catégorie de l'individu x. K est le nombre d'individus dans la catégorie considérée et $d(x,y)$ la distance basée sur les mesures à optimiser, entre

les individus x et y. La distance *disInf* est la distance d'influence et définie le rayon de la niche écologique de la catégorie. L'objectif est de déterminer la meilleure approximation de l'ensemble des solutions de Pareto, ou, du moins une approximation satisfaisante. Idéalement, cet ensemble doit être contenu dans la population finale de l'AG à l'issue de son exécution. Néanmoins, ce résultat n'est pas garanti. En effet, la nature stochastique de l'AG peut provoquer la disparition via l'opérateur de mutation, d'individus non dominés. D'autre part, l'étendue de la frontière de Pareto n'est pas forcément connue a priori et, de fait, le dimensionnement de la population en conséquence pose problème. Ces deux phénomènes conduisent à utiliser une archive des solutions non dominées trouvées. Lorsqu'une telle solution est rencontrée suite au croisement ou à la mutation, elle est stockée dans l'archive avec éventuellement une mise à jour de celle-ci. Ainsi, les solutions présentées à l'utilisateur sont celles présentes dans l'archive et non celles de la population finale. L'algorithme NSGA a été apliqué à notre problème sur les jeux de données *Vote, Sonar et Churn* pour la recherche de règles optimisant les critères présentant un antagonisme apparent.

- **Résultats**

Les résultats obtenus font apparaître l'intérêt de l'AG dans le parcours de l'espace des tailles de règles, dans le parcours de l'espace des règles muni de la distance définie au paragraphe I.3.3 et dans la sélection de règles selon des critères multiples. Nous reprenons chacun de ces points dans ce paragraphe.

- **Taille des règles**

Notons que la représentation que nous avons choisie permet justement de représenter toutes les longueurs puisque chaque gène code, en particulier, la présence ou l'absence de l'attribut correspondant dans la règle. L'observation de l'espace de recherche présentée au paragraphe II.3.3 montre une variété importante dans la taille des règles optimales selon un critère donné. Sur un même jeu de données, on peut trouver des règles de petite taille aussi bien que des règles de taille importante. Les règles longues sont en général ignorées par les algorithmes standards. Par exemple, un algorithme d'induction d'arbre de décision passe par une phase d'élagage tendant à réduire la longueur des branches. D'un autre côté, ces règles longues couvrent généralement peu d'exemples et ont un support faible ; elles ne sont pas découvertes par un algorithme de recherche de règles d'association. C'est d'ailleurs pour cette raison que ces règles sont potentiellement intéressantes, car elles peuvent représenter un événement rare ou précis dont la connaissance est instructive pour l'utilisateur. Il est

donc important que la méthode de recherche n'introduise pas un biais tendant à favoriser certaines tailles. Les règles longues qui ne sont qu'une version redondante de règles courtes découvertes ne présentent aucun intérêt ; dans le cas contraire, elles méritent d'être sélectionnées si elles sont différentes des règles courtes, car elles peuvent traduire une corrélation très précise et non soupçonnée. Par exemple, l'AG que nous avons utilisé, extrait la règle donnée ci-dessous sur *Vote* . Elle présente 3 attributs dans l'antécédent et implique une conjonction de 7 termes attribut-valeur avec une mesure de confiance élevée. Cette règle maximise la *précision*, mais sa valeur de *support* (12.4%) est faible, elle ne serait déterminée par un algorithme de type APriori qu'en baissant le seuil de *support* minimum.

si immigration=y ET crime=y ET class=republican

alors physician-fee-freeze=y ET el-salvador-aid=y ET religious-groups-in-schools=y ET aid-to-nicaraguan-contras=n ET mx-missile=n ET superfund-right-to-sue=y ET duty-free-exports=n

- **Recherche locale versus recherche globale**

En ce qui concerne le parcours de l'espace de recherche, nous avons vu que les algorithmes classiques sont basés sur une exploration locale. La Figure II.4 illustre cette situation ; elle met en évidence l'existence de règles meilleures, pour la *précision*, que la règle extraite par une recherche locale. On a montré, dans ce cas que l'AG cherche ailleurs dans une autre région de l'espace et trouve de meilleures règles et éloignées d'un optimum local selon la distance phénotypique que nous avons définie en II.3.3. En effet, la notion de localité induite par cette distance correspond à la distance entre un chromosome et un de ses descendants obtenus par application des opérateurs génétiques décrits ci-dessus ; l'AG démontre là sa capacité à s'échapper d'un optimum local.

- **Multi-critères**

La qualité d'une règle s'exprime de manière multiple ; la variété des indices d'intérêt le montre bien. On peut en effet considérer qu'une règle est intéressante parce qu'elle couvre de nombreux exemples et qu'elle traduit une différence par rapport à la situation où son antécédent et son conséquent seraient indépendants. Dans ce cas, on va chercher à optimiser simultanément *support* et *confiance*, ou *support* et *conviction*. On pourrait également ajouter une contrainte sur la taille.

La Figure II-9 montre le résultat de l'algorithme NSGA mis en œuvre sur ce problème multi-critères. Les triangles représentent les règles trouvées par l'AG. En effet, sur la Figure II-5, on observait, par exemple, l'antagonisme apparent des indices de *support* et de *conviction* sur *Vote*. L'algorithme

multiobjectifs confirme cette propriété et permet de découvrir un ensemble de règles sur la frontière de Pareto.

En ce qui concerne les indices de *support* et *conviction* (ou *confiance*) qui sont à la base de l'extraction de règles d'association, l'AG multiobjectifs permet d'obtenir directement un ensemble de règles optimisant l'ensemble des critères simultanément. On cerne ainsi directement un ensemble de compromis qu'une méthode de type APriori obtiendrait laborieusement en procédant par une succession d'essais et de tests et en abaissant les seuils minimaux jusqu'à obtenir des résultats jugés satisfaisants. Notons de plus que cette méthode produirait une masse de règles candidate elle-même à être fouillée!

Une différence importante avec les algorithmes standard réside dans le fait que l'AG est indépendant des critères de qualité à optimiser et peut être facilement adapté à tout indice et à toute combinaison d'indices. Cette souplesse a permis de tester la recherche de règles optimales selon d'autres ensembles de critères.

Nous avons ainsi pu utilisé cet algorithme pour rechercher les meilleurs compromis selon la *sensibilité*, la *spécificité* et la *longueur*. La combinaison des deux premiers critères permet de quantifier un autre aspect de la qualité d'une règle ; elle permet de trouver des règles fortes exprimant une condition nécessaire et suffisante pour que le conséquent soit réalisé.

Sur la table *Vote*, l'algorithme multiobjectifs découvre un ensemble de règles sur la frontière de Pareto. Par exemple, nous avons obtenu la règle inédite suivante :

si *physician-fee-freeze=yes* ***et*** *el-salvador-aid=yes* ***et*** *duty-free-exports=no*

alors *classePolitique=republican*

avec une sensibilité de 0.86 et une spécificité de 0.97 ; cette règle peut être vue comme l'expression d'une bonne caractérisation de la classe *republican* dans la mesure où environ 86% des exemples (vrais positifs) de cette classe et moins de 4% des exemples (faux positifs) n'appartenant pas à cette classe vérifient l'antécedent de la règle.

II.5. Conclusion

Dans ce chapitre, nous avons présenté nos travaux en cours sur la recherche des meilleurs motifs en fouille de données. Bien que cette problématique s'exprime naturellement en termes d'optimisation, ce

point de vue est peu considéré dans le cadre de la fouille de données. C'est ce constat qui nous a motivé à traiter la recherche de motifs comme un problème d'optimisation.

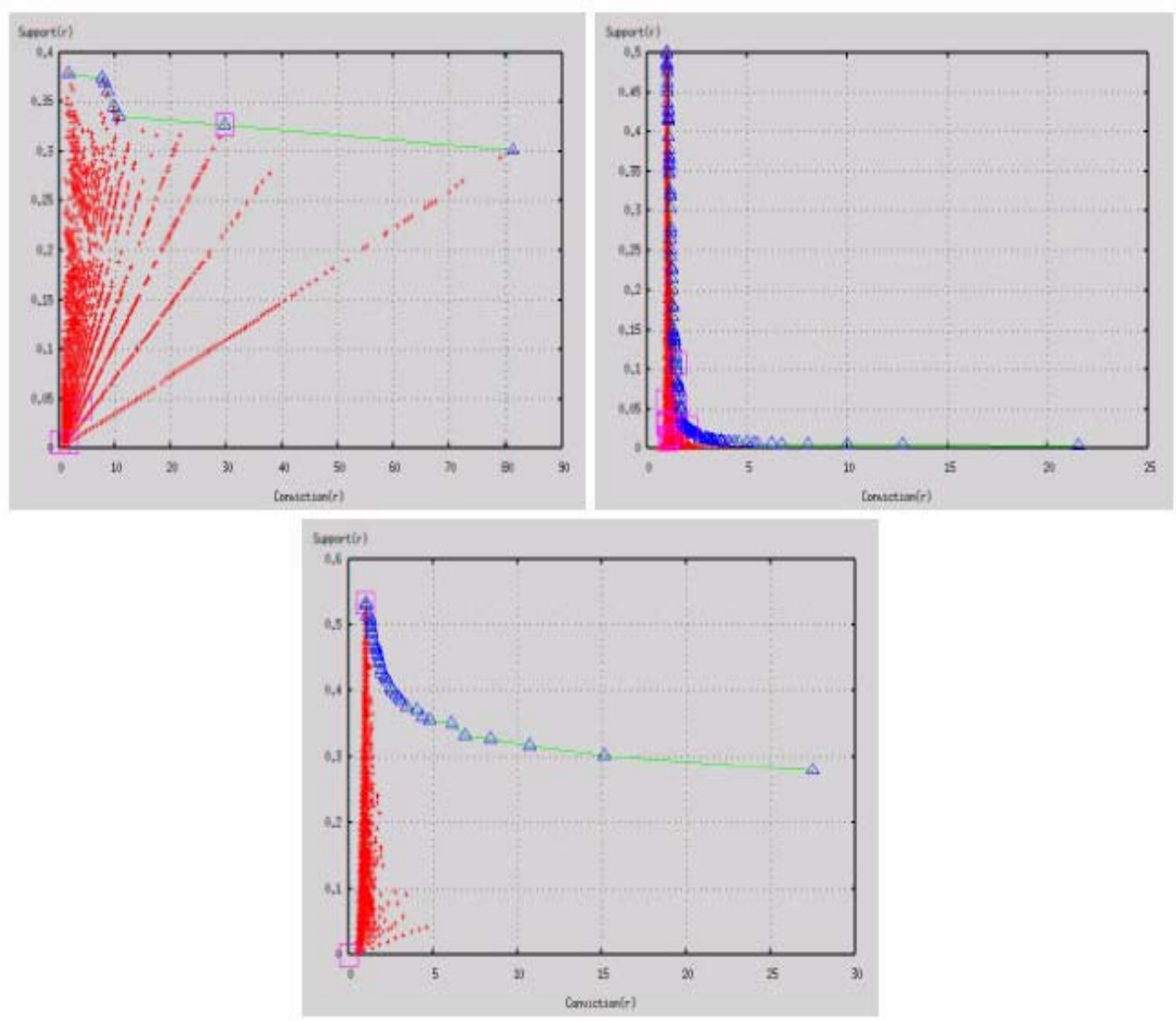


Figure II-9 Optimisation multi-critères s*upport* et *conviction*

Ceci nous a amené à étudier les caractéristiques des espaces de recherche, puis, à déterminer en quoi certains algorithmes sont plus adaptés. Nous avons traité non seulement de méthodes d'optimisation, mais également de mesures d'évaluation des motifs et des approches évolutionnaires en fouille de données.

En effet, nous avons considéré les motifs à base de règles qui sont très compréhensibles. Ces travaux, qui sont menés actuellement dans le cadre d'une thèse [FRAN04], ont consisté, en premier lieu, à considérer l'ensemble des règles associées à leurs indices de qualité comme espace de recherche. Parmi le large éventail d'indices de qualité, nous avons retenu des mesures de base, ayant un sens

concret et facilement interprétables qui ont été définies comme mesures d'*intérêt*. Nous avons traité le cas de l'optimisation individuelle de chaque indice, ainsi que l'optimisation simultanée de plusieurs indices. L'espace de recherche a été analysé en étudiant la densité d'états des indices, la variation de leurs valeurs en fonction de la taille des règles, et la variation de leurs valeurs en fonction de la distance des règles à une règle optimale donnée. Enfin, nous avons analysé les variations relatives des indices pris deux à deux.

Les propriétés des espaces de recherche (combinatoire élevée, existence de minima locaux, indices antagonistes) justifient le recours à des algorithmes évolutionnaires. Nous avons effectivement montré que, dans certains cas, ces algorithmes permettaient de découvrir des motifs ignorés par d'autres approches. Toutefois ces résultats ne démontrent pas l'avantage des algorithmes évolutionnaires sur une méthode traditionnelle, dans toutes les situations de recherche de motifs.

Aussi, le bilan de cette étude est essentiellement d'ordre méthodologique ; il suggère que la recherche des meilleures règles débute en identifiant clairement le sens donné à la qualité, l'intérêt ou l'utilité des règles et les indices choisis pour évaluer ces critères, puis en étudiant l'espace de recherche produits par ces indices.

Ces résultats permettent d'améliorer la qualité des règles sélectionnées selon des critères d'intérêt objectifs. Cependant l'application de ces nouveaux principes peut produire, comme les méthodes standard, un volume de règles très important qui nécessite un nouveau tri selon des critères subjectif prenant en compte l'utilité pour l'utilisateur final.

Chapitre III Fouille de données

Contribution applicative

III.1. Introduction

Dans la plupart des organisations actuelles, les systèmes d'information (SI) automatisés constituent un avantage compétitif. Ils jouent un rôle central dans la vie de l'entreprise et fournissent des outils indispensables à un fonctionnement adapté à ses activités ordinaires. Le fonctionnement des

SI repose sur le développement d'applications en général couplées à des bases de données qui permettent de stocker en mémoire persistante des informations manipulées et interrogées par les programmes. Ces bases, véritables réservoirs d'information, contiennent des données relatives aux champs d'activité très variés de l'entreprise. Leur volume est en croissance permanente du fait des progrès des performances techniques des systèmes de saisie et de stockage, d'une part, et des mises à jour imposées par les évolutions fréquentes des activités, d'autre part. Désignés, par les anglo-saxons, sous le terme de "*legacy data bases*", l'état de ces bases de données *anciennes*, conduit inéluctablement à opérer un changement radical.

Une décision essentielle pour l'analyste porte sur le choix d'exploiter les structures existantes pour opérer la transformation ou de procéder à une conception ex-nihilo qui ignore les précédents travaux d'analyse et de conception. La solution consistant à exploiter les systèmes existants permet de tirer avantage des enseignements donnés par la construction initiale, en recherchant toutes les informations sur sa conception et ses fonctionnalités. Un autre argument en faveur de cette solution réside dans l'observation des bases de données anciennes, pour lesquelles il n'existe pas de documentation produite après la conception initiale du système et dont les premiers concepteurs ne sont plus présents ; dans ce genre de situations, l'étude du système est plus instructive qu'une conception ignorant l'existant.

Chikofsky et Cross [CHIK90] ont défini la rétro-conception d'un système comme "*l'analyse du système existant de manière à identifier ses composants et les relations qui les lient, et à construire une représentation du système sous une autre forme ou à un niveau d'abstraction plus élevé*". La plupart des travaux de recherche sur le thème de la rétro-conception de bases de données (RCBD) consiste à examiner le système existant et à en extraire une représentation abstraite. L'acceptation courante du processus de rétro-conception est conforme à la définition de Chikofsky et Cross ; elle ne prend pas en compte la création d'un nouveau système. La phase de rétro-conception correspond à une étape essentielle dans le processus d'évolution des bases de données anciennes. C'est, en effet, lors de cette phase, que sont étudiées les possibilités d'évolution vers de nouvelles fonctionnalités et de nouvelles structures aptes à supporter de nouvelles activités de l'entreprise. Par exemple, l'évolution vers des modèles de données plus complexes et plus puissants, est un point central dans la mise au point de techniques de re-structuration et de rétro-conception.

Dans le cas particulier des bases de données, la rétro-conception consiste à "comprendre" la base de données ancienne pour retrouver les spécifications de conception initiales et faciliter une nouvelle implémentation. Cette tâche est compliquée par le fait que les bases de données anciennes, ont subi de

nombreuses modifications sans référence à un modèle sémantique donné ; de plus, les indications explicites sur les domaines sémantiques sont souvent inexistantes.

Un processus de RCBD consiste en l'application de méthodes permettant de comprendre la structure et la sémantique des données et de réorganiser et de construire un nouveau schéma conceptuel. Les principaux objectifs sont :

- traduire la structure de la base de données dans une forme plus compréhensible par l'utilisateur,
- "nettoyer" le schéma existant des différentes scories introduites quelques fois volontairement pour optimiser le fonctionnement,
- extraire des informations structurelles et sémantiques sur les données et les représenter dans un modèle conceptuel.

Le nettoyage des données est particulièrement sensible dans des bases de données en usage depuis de nombreuses années, multiples et hétérogènes. Par exemple, dans le domaine bancaire, sont utilisées traditionnellement des bases séparées pour chaque champ d'activité, comptes courant, crédits logement, crédits à la consommation, etc. Pour obtenir un description complète du système, par exemple pour explorer la relation avec les clients, il est nécessaire de couvrir l'ensemble des données. Du fait de l'existence de base de données anciennes émanant de plusieurs services, on trouve par exemple des définitions différentes et des ensembles de valeurs différents pour un même champ. L'ignorance de ce type d'anomalie conduit à des incohérences et des interprétations peu fiables. La violation de contraintes d'intégrité constitue une autre source de confusion. Le contrôle de l'intégrité référentielle est un des atouts des bases de données relationnelles qui a permis d'améliorer la qualité de l'information stockée. Mais les bases de données existantes, construites dans des systèmes peu contraignants, sont souvent incomplètes et renferment de nombreuses erreurs.

Les sources d'information utiles pour la RCBD sont multiples : généralement, le schéma de la base de données et les données elles-mêmes sont analysés, mais les applications et les requêtes donnent également des informations importantes. Le modèle de bases de données actuellement le plus utilisé est le modèle relationnel, défini dans les années 1970. Bien que l'approche objet s'impose dans le contexte de la programmation, la plupart des environnements orientés objet, comme les environnements de programmation Java offrent des interfaces vers la majorité des types industrialisés de bases de données relationnelles. Les travaux actuels de rétro-conception se focalisent donc sur des bases de données de type relationnel.

La méthode EXORE (EXtraction de schémas Objet pour la REtro-conception de bases de données) présentée dans ce mémoire, répond à différents objectifs motivés par l'observation des bases

de données actuelles et des méthodes proposées. De manière générale, EXORE vise plusieurs objectifs :

- offrir une approche réaliste pour des bases de données en usage, éventuellement dégradées,
- exploiter le volume important des informations de toutes sortes, stockées dans ces bases anciennes, par des techniques appropriées issues du domaine de la fouille de données ; les cas des bases de données réelles, utilisées dans l'industrie, exhibent de nombreuses déviations par rapport aux modèles définis théoriquement et souffrent également d'erreurs de conception,
- se baser sur un modèle de données cible, proche des modèles logiques actuellement utilisés, relationnels et orientés objet. En effet, les bases de données actuelles susceptibles de nécessiter une reconstruction, sont de type relationnel, mais la dernière version du langage SQL (SQL3) montre la nécessité d'évoluer vers un modèle plus complexe, alliant les avantages du relationnel et de l'objet. Dans ce contexte, il semble important, pour une méthode de rétro-conception de faciliter une implémentation conforme aux tendances actuelles plutôt que de rechercher inutilement un niveau d'abstraction générique,
- distinguer deux phases dans le processus de rétro-conception : une phase d'analyse et de découverte, suivie d'une phase de reconstruction. La phase d'analyse, initiale permet non seulement d'établir un catalogue des structures et informations explicitement déclarées, mais également de mettre à jour l'information implicitement stockée dans la base. La découverte des anomalies de nommage, du non-respect des contraintes d'intégrité, de structures optimisées, etc. intervient à ce niveau et alimente la phase de reconstruction qui identifie les éléments du schéma décrivant la base existante.

La suite du chapitre est organisée de la manière suivante :

- la section III.2 présente un état de l'art synthétique des travaux de rétro-conception de bases de données,
- la section III.3 est consacrée à la présentation des principes de la méthode EXORE et du modèle de données cible associé MORE.

- la section III.4 présente l'approche "fouille de données" mise en œuvre dans la phase d'analyse d'EXORE,
- la section III.5 conclut et présente nos projets de travaux.

III.2. Méthodes de rétro-conception

Les méthodes de RCBD ont naturellement suivi l'évolution des bases de données : les premières méthodes ont concerné les structures COBOL, puis les étapes suivantes ont vu le développement de méthodes permettant de restructurer des bases de données réseau ou hiérarchiques. Les publications récentes sur ce domaine de recherche se focalisent sur les bases de données relationnelles. Les résultats présentés sont très divers, tant du point de vue du spectre couvert, que des types de sujets traités. On peut distinguer les publications faisant état de certaines expériences de rétro-conception et qui exposent, en général, des problèmes posés par le caractère incomplet, erroné, dégradé des bases existantes. D'autre part, on regroupe des publications plus théoriques, se focalisant sur des problèmes formels et qui supposent souvent de disposer de structures nettoyées de toute anomalie. Chaque méthode de rétro-conception a ses propres caractéristiques méthodologiques, utilise des sources d'information plus ou moins variées et suppose qu'un certain nombre de conditions soit vérifié par la base de données source. Ces méthodes poursuivent le même objectif de renforcer l'automatisation du processus de rétro-conception ; néanmoins, dans tous les cas, une part d'interactivité avec l'utilisateur, expert du domaine, est pour l'instant nécessaire.

Ainsi, M. Blaha [BLAH01] présente un exemple de rétro-conception d'une base de données industrielle et met en évidence différentes anomalies de conception détectées. Il organise la rétro-conception en trois phases qui se déroulent dans l'ordre inverse du processus de conception : l'étude de l'implémentation qui permet de représenter chaque table comme une entité, puis, l'étude de la structure de la base qui permet de résoudre les clés primaires, et enfin, l'interprétation du modèle obtenu, son nettoyage et sa restructuration. Dans cette expérience, le degré d'automatisation est assez réduit : en comparant les types de données et les contraintes sur deux attributs, l'analyste a l'intuition que deux attributs sont similaires et en demande confirmation aux développeurs. De même, les clés étrangères sont repérées manuellement d'après leur nom, si elles ne sont pas déclarées explicitement dans le

schéma. La méthode appliquée dans cette expérience, procède de manière assez classique en se basant sur le nom des attributs, pré-supposant la pertinence du nommage.

On peut classer les différentes méthodes de RCBD selon de multiples critères : le modèle de la base de données source, le modèle conceptuel cible, les différentes sources d'information utilisées (données, schéma, commentaires, programmes, requêtes, …) et la manière d'exploiter ces informations. Une caractéristique qui apparaît essentielle pour évaluer l'intérêt d'une méthode réside dans les contraintes qu'elle suppose être vérifiées par la base de données source. En effet, il est peu réaliste de définir une méthode fonctionnant uniquement sur une base de données sans anomalie, normalisée et cohérente. Nombreuses sont les méthodes proposées qui se fondent sur des hypothèses fortes sur l'intégrité des bases de données. Moins de travaux, par exemple, s'intéressent à la dénormalisation, à la "désoptimisation" de la base de données ou aux problèmes d'incohérence dans le nommage des attributs.

Le modèle cible, support du schéma conceptuel résultant du processus revêt également une importance particulière. Il est, dans la plupart des cas, d'un haut niveau d'abstraction. Le processus est guidé par le schéma physique existant ou plusieurs schémas physiques en cas d'intégration de sources de données multiples. Il est, en fait, difficile, sinon impossible de retrouver le schéma conceptuel d'origine car certains choix d'implémentation ne sont pas commentés et les concepteurs initiaux n'ont laissé aucune trace de leur réflexion. Donc, les processus de RCBD corrigent les anomalies, retrouvent des liens cachés et des indices permettant d'exprimer des conjectures sur les motivations des concepteurs mais ne remettent pas en cause tous les choix d'implémentation initiaux.

Les techniques utilisées pour découvrir les informations nécessaires à la construction d'un schéma conceptuel, sont souvent qualifiées de techniques d'extraction, faisant une référence implicite au domaine de *l'extraction automatique de connaissances à partir des données*. Cependant, peu de méthodes font réellement état de la mise en œuvre des techniques de *fouille de données*. On peut citer, comme exemple, un des rares travaux utilisant ces techniques : ceux de I. Comyn-Wattiau et J. Akoka [WATT96] qui mettent en évidence des structures dénormalisées et optimisées par extraction.

La méthode proposée par R. Chiang [CHIA94] est décrite en détail ; elle montre comment on peut retrouver, dans un schéma existant, les composants et les liens entre composants issus des implémentations passées. Les publications récentes dans ce domaine se focalisent plutôt sur un problème précis : l'identification de structures optimisées [WATT96], la recherche de liens complexes mis en place dans des structures relationnelles, l'analyse de requêtes SQL [PETI96], la recherche de

dépendances fonctionnelles. Quelques études font état d'expériences pratiques et présentent des synthèses des différents cas de figures observés [PREM94, AIKE99].
La plupart des approches pré-supposent, en général, qu'un ensemble de conditions favorables et en général restrictives, peu réalistes soient vérifiées par la base de données existante. De même, la base de données est supposée en troisième forme normale et plus généralement, les possibilités d'optimisation de la base de données, pour des raisons de performance sont ignorées. Une contrainte de départ qui est assez importante concerne la politique de nommage des objets (attributs, relations, …) qui est supposée cohérente et fiable.
D'autre part, ces méthodes essaient d'extraire un modèle conceptuel alors que l'existant ne permet pas toujours de retrouver, de manière exhaustive, l'analyse des premiers concepteurs. Ainsi, les phases d'analyse et d'abstraction du nouveau schéma sont souvent confondues, ce qui fait que l'exploitation des sources d'information utiles n'est pas réellement approfondie.

III.3. Méthode EXORE : Principes et modèle de données

Dans des organisations dont les systèmes d'information automatisés reposent sur la manipulation de bases de données anciennes et obsolètes, la rétro-conception de bases de données permet de ne pas remettre en cause tous les choix de conception des applications et favorise la migration des données vers de nouvelles structures plus adaptées. Si cette approche permet d'obtenir des résultats de meilleure qualité exploitant l'analyse et la conception initiales, l'état réel des bases de données opérationnelles en accroît la difficulté. La méthode EXORE (EXtraction de schémas Objet pour la REtro-conception) présentée dans ce chapitre définit une approche adaptée aux bases de données opérationnelles ne répondant pas toujours aux normes théoriques.

III.3.1. Objectifs

En définissant la méthode EXORE, nous avons recherché une approche réaliste, sans a priori accommodant sur l'état des données, n'ignorant pas la présence d'anomalies et d'incohérences inhérentes au fonctionnement d'un base de données en exploitation. Car, dans le cas des bases de données réelles, utilisées dans l'industrie, on observe de nombreuses déviations par rapport aux modèles décrits en théorie. Actuellement, les bases de données sujettes à un processus de

reconstruction, sont pour leur grande majorité relationnelles. En effet, malgré des capacités beaucoup plus étendues de modélisation des données, les systèmes de gestion de bases de données orientées-objet (SGBDOO) n'ont pas réussi, pour l'instant, à supplanter les systèmes de gestion de bases de données relationnelles (SGBDR) dont la simplicité du modèle sous-jacent fait également l'efficacité. La théorie des bases de données relationnelles définit des normes sur les structures et les données, qui ne sont pas nécessairement respectées dans les bases de données industrielles. Les déviations par rapport au modèle théorique sont dues à des erreurs de conception et d'implémentation, à une mésestimation de l'importance des normes pour la cohérence du système et, enfin, à des impératifs de performance dans l'exécution des transactions. La difficulté de la tâche de reconstruction se trouve nettement accrue devant ces structures non normalisées et la prise en compte de ces aspects doit diriger le processus d'analyse de la base de données existante. Dans EXORE, nous nous intéressons plus particulièrement aux anomalies de nommage. Les différents cas d'anomalies sont détaillés dans la section III.3.2.

Le second objectif est d'exploiter le volume souvent très important des données existantes. Les bases de données anciennes, désignées sous le terme de *legacy databases* par les anglo-saxons, sont soumises à de nombreuses mises à jour et sont, dans certains cas, augmentées quotidiennement de milliers d'enregistrements. Aussi, une caractéristique notable de ces bases réside dans leur important volume. Par exemple, la Tableau III-1 donne un aperçu de la taille des bases manipulées lors d'une expérience de rétro-conception ; les base ont été utilisées pour la gestion de la paie et du personnel dans une université de Virginie [AIKE99]. On pourrait affirmer que ces volumes suggèrent naturellement de recourir à des techniques de fouille de données ; cependant les approches de ce type sont encore rares en rétro-conception. Nous avons introduit, dans EXORE, des techniques de fouille utiles à l'analyse de la base existante ; elles sont présentées dans la section III.4.

Système	Age	Nombre d'enregistrements	Nombre d'entités	Nombre d'attributs
Paie	15 ans	780 000	35	683
Personnel	21 ans	4 950 000	57	1478

Tableau III-1 Exemples de Bases de données

III.3.2. Bases de données opérationnelles

Les bases de données opérationnelles sont soumises à des contraintes de performance qui conduisent les concepteurs à définir des structures optimisées, non normalisées. Pour des raisons de performance dans l'exécution des transactions, l'implémentation physique du schéma logique normalisé de la base de données n'est en général pas la représentation la plus efficace. Ainsi, des opérations d'éclatement et de dénormalisation sont effectuées sur le schéma logique pour éviter le calcul de jointures trop coûteuses ou pour réduire la taille des données accédées le plus fréquemment. Ce type d'anomalies introduites par des relations qui ne sont pas en 3NF se traduit par la présence de dépendances fonctionnelles qui ne sont pas directement dépendantes de la clé. Dans un processus de rétro-conception, il est fondamental de pouvoir connaître la totalité des dépendances fonctionnelles et non seulement celles que l'on retrouve à partir des clés. Des algorithmes [MANN87, SAVN93] dont le plus connu est TANE [HUHT99] ont été proposés pour découvrir des dépendances fonctionnelles à partir de l'analyse des données. De même, toujours pour des raisons de performance, le schéma de la base de données peut subir des opérations de structuration tel que l'éclatement ou l'aplatissement de certaines relations comme le montre I. C. Wattiau et al. [WATT96]. Ces opérations préservent en général les formes normales, mais séparent une partie des données de façon à minimiser le temps d'accès pour les programmes d'application.

L'éclatement de relations est, soit horizontal car il permet d'isoler un ensemble de tuples, soit vertical car il permet d'isoler un ensemble d'attributs. Dans EXORE, l'identification de ces types d'éclatements ainsi que leurs conséquences sur le schéma intervient dans la phase d'analyse décrite plus loin.

Les méthodes de rétro-conception font reposer une partie de leur analyse et de l'identification des entités sur les noms des attributs dans la base de données existante. Cependant cette approche peut conduire à des conclusions erronées ; le nommage des attributs est un critère trop faible pour fonder l'analyse des structures existantes dans la mesure où le nom d'un attribut n'identifie pas de manière certaine le concept du monde réel qu'il représente. La stratégie de choix des noms d'attributs et de relations n'est pas toujours clairement définie par les concepteurs initiaux d'une base de données. De plus, la phase de maintenance introduit au cours du temps, des anomalies terminologiques comme des cas de synonymie et d'homonymie. Dans ce contexte, des attributs synonymes représentent le même concept du monde réel et ont été affectés d'identificateurs différents lors d'opérations de mise à jour des structures ou d'intégration de données, par exemple. De même, on trouve des attributs homonymes dont les identificateurs sont des termes génériques et donc imprécis comme `nom`, `numéro`,

`libellé`. La difficulté de la tâche d'analyse consiste alors à découvrir que ces attributs de même nom représentent en fait des concepts différents du monde réel.

L'origine de ce problème vient de la prise en compte de la notion de domaine sémantique défini par E. Codd dans la théorie des bases de données relationnelles [CODD70]. Les SGBD relationnels ne permettent pas, en général, d'implémenter la notion de domaine sémantique. Cependant, le concept de domaine joue un rôle important dans la sémantique attachée aux données relationnelles. Il est l'identificateur de la sémantique attachée à l'attribut et à ce titre évite toute confusion entre attributs. Ce concept dépasse la notion de type affecté à une variable dans les environnements de programmation. Il permet en particulier d'identifier une clé étrangère sans déclaration explicite. Dans la grande majorité des SGBD commercialisés, le concept de domaine est réduit en étant traduit par le concept plus générique de type de données. Les liens entre tables doivent être définis de manière explicite en déclarant des attributs comme clé étrangère. On observe ainsi des situations dans lesquelles les concepteurs de la base de données n'ont pas réalisé une construction rationnelle des structures dans lesquelles, par exemple, certains attributs qui devraient être de même domaine sémantique ne sont pas déclarés comme tels [PREM94, BLAH98a].

En effet, les motivations qui animent les concepteurs sont souvent dirigées par des contraintes imposées par le contexte de mise en œuvre de la base : SGBD, langages de programmation, langages de quatrième génération … qui utilisent les données. On peut ainsi voir un attribut qui tient le rôle de clé étrangère et qui n'est pas déclaré de même type que la clé primaire référencée. Dans certains cas, le nom de l'attribut suffit à l'identifier ; c'est le postulat que font certaines méthodes de rétro-conception, mais rien n'assure que cette politique de nommage soit respectée. Les opérations d'optimisation peuvent également aboutir à ce type d'anomalie. Il n'est donc pas réaliste de baser le processus d'analyse du schéma existant sur l'observation des noms d'attributs. Comme nous l'avons vu dans le chapitre II, certaines méthodes de rétro-conception ignorent ce problème en supposant que des attributs de même domaine, sont dénommés de la même manière. On peut en outre, observer fréquemment des attributs homonymes dont les noms sont plutôt génériques et ne permettent pas de conclure à une similarité.

D'autres méthodes examinent non seulement les noms d'attributs, mais leur ensemble de valeurs. Ce critère apporte un élément de fiabilité, mais peut être mis en défaut ; on peut par exemple avoir deux champs `R1.numero` et `R2.numero` ayant des valeurs identiques et représentant malgré tout des concepts différents.

Des méthodes plus sophistiquées [ANDE94, PETI96] identifient des attributs de même domaine en recherchant des équi-jointures entre attributs. Cette approche se base sur la conjecture selon laquelle des attributs de même domaine sont nécessairement utilisés dans des équi-jointures. On observe cependant, dans les bases de données en usage depuis de longues années, des situations qui ne permettent pas de conclure aussi rapidement. L'objectif d'efficacité dans les performances de réponse aux requêtes est souvent prépondérant au détriment des principes de respect d'intégrité. Les structures sont le plus souvent pas en forme normale et optimisées de manière à réduire le nombre de jointures. On peut voir co-exister des attributs synonymes avec une probabilité faible d'occurrence d'une équi-jointure sur ces attributs. Les situations résultant de l'intégration de plusieurs sources de données peuvent également aboutir à des anomalies de nommage. Par exemple, on peut observer des noms de relations et d'attributs synonymes exprimés par des termes anglais et français ; ce cas est souvent rencontré dans le cas d'intégration de schémas lors d'une fusion de différentes sources de données. Ce cas peut se produire lors de l'intégration de schéma par exemple, d'où la nécessité de savoir que les attributs représentent les mêmes informations et que c'est peut-être la même table.

Pour une analyse fiable, il est cependant essentiel de mettre à jour ses similarités ou dissimilarités d'attributs pour identifier des structures elles-mêmes similaires ou des liens particuliers entre structures. La technique définie dans EXORE pour identifier les similarités est basée sur un algorithme de calcul de distance contextuelle détaillé à la section III.4. L'extraction de liens de similarité est réalisée en exploitant le volume de données selon une approche de type *fouille de données*. Il est effectivement fondamental de fournir des outils d'automatisation aidant l'utilisateur dans la phase d'analyse, qui peut difficilement être réalisée manuellement au regard du volume des données.

La manière dont sont implémentés certains liens sémantiques est également source de difficulté pour la rétro-conception. Les liens d'héritage qui peuvent apparaître dans un schéma conceptuel ne se traduisent pas directement dans un schéma relationnel. Il en est de même pour les attributs multivalués. Le modèle relationnel, le plus répandu dans les bases de données en usage actuellement, repose sur des structures simples et ne fournit pas de concept pour traduire des liens sémantiques complexes comme la spécialisation/généralisation entre entités. Les schémas conceptuels incluant des liens d'héritage sont implémentés "au mieux" dans des schémas logiques relationnels en détournant le rôle premier des clés primaires et des autres attributs. On peut distinguer deux techniques pour implémenter un lien de généralisation/spécialisation entre entités. La première consiste à utiliser les valeurs nulles pour introduire deux types de données dans une même relation. La seconde technique utilise des attributs clés primaires communs pour identifier les instances d'un type et d'un sous-type.

Nous présentons ces deux approches à partir d'exemples. Pour le processus de rétro-conception, il s'agit donc d'identifier ces cas de figure.

Les attributs multivalués ne sont pas pris en compte tels quels dans un schéma relationnel en première forme normale. Prenons l'exemple du concept d'`Ouvrage` pour lequel on peut avoir plusieurs auteurs. Par souci de simplicité et pour éviter des jointures coûteuses, le programmeur peut modéliser cet attribut-collection en créant plusieurs attributs de même domaine : $\texttt{auteur}_1$, $\texttt{auteur}_2$, … $\texttt{auteur}_n$. Selon cette technique, si chaque ouvrage peut avoir jusqu'à quatre auteurs, la relation `Ouvrage` a la structure suivante : `Ouvrage(NumOuvrage, Editeur, Auteur1, Auteur2, Auteur3, Auteur4, NbrePages)`. Une autre technique consiste également à voir le lien comme une agrégation et à créer une nouvelle relation avec une clé multi-attribut, par exemple la relation `AuteurOuvrage (NumArticle, NumOuvrage)` est construite de cette manière. Le processus de rétro-conception doit détecter ces attributs qui peuvent être représentés en tant que collection dans un modèle objet.

III.3.3. Modèle de données MORE

La méthode EXORE (**EX**traction de schémas **O**bjet pour la **RE**tro-conception) est conçue de manière à proposer une alternative aux approches classiques de rétro-conception en proposant d'extraire, non pas un schéma conceptuel de type *Entité-Association* ou *Entité-Association Etendu*, mais un schéma d'un niveau d'abstraction intermédiaire, exprimé selon le modèle MORE (**M**odèle **O**bjet pour la **RE**tro-conception) proche des modèles logiques actuellement utilisés.

Nous nous plaçons dans le contexte exclusif des bases de données relationnelles pour ce qui concerne les données existantes. Cependant, nous ne pouvons ignorer le modèle objet qui offre des possibilités de modélisation beaucoup plus étendue et permet la définition de types de données complexes alors que le modèle relationnel se limite à des données plates. Les SGBDOO n'ont pas réussi à s'imposer pour l'instant en grande partie, parce qu'ils ne supportent pas bien le traitement efficace de requêtes complexes. Mais les structures relationnelles, elles, présentent l'inconvénient de limiter les possibilités de modélisation. Par exemple, pour traduire un lien de généralisation, le programmeur doit détourner les moyens fournis (table, clés, valeurs nulles) pour prendre en compte ces liens complexes. Ainsi, l'objectif principal de rétro-conception de données relationnelles est de fournir une description plus abstraite des données permettant à la fois une meilleure adéquation au domaine d'application et des possibilités d'extension. Dans EXORE, le processus fournissant le schéma cible se base sur le modèle

intermédiaire MORE permettant de bénéficier simultanément des avantages du relationnel et de l'objet. Nos objectifs sont les suivants :

- réduire la difficulté du processus d'abstraction vers un modèle conceptuel en ciblant plutôt un modèle intermédiaire entre modèle conceptuel et modèle logique,
- rester proche des modèles logiques actuellement utilisés et dont le spectre est assez étroit : relationnel, objet et relationnel-objet,
- permettre d'intégrer le processus de rétro-conception des données dans le processus général de re-conception du système,
- conserver, au niveau de ce modèle, les concepts de structures simples qui caractérisent le relationnel et des concepts plus puissants permettant de représenter des structures complexes et d'encapsuler les données et les opérations sur ces données,
- faciliter la migration future des données vers le modèle logique objet-relationnel qui constitue un bon compromis pour représenter des objets plus complexes et exécuter des requêtes complexes sur des structures simples.

Le modèle MORE est basé sur les concepts d'*entité-classe* et d'*entité-relation*. Il est naturellement proche du modèle objet-relationnel, mais reste cependant plus abstrait puisque son rôle est de fournir une description synthétique des donner existantes.

La notion de classe issue de l'approche objet permet de décrire le domaine de définition d'un ensemble d'objets. Selon l'approche objet, chaque objet appartient à une classe. Les généralités sont contenues dans la classe et les particularités sont contenues dans les objets. Les objets sont construits à partir de la classe, par un processus d'instanciation. De ce fait, tout objet est une instance de classe. La classe est une description d'une famille d'objets similaires possédant un état, décrit par une structure constituée d'attributs, et un comportement, décrit par des opérations appelées méthodes. Une instance d'objet est une unité atomique formée de l'union d'un état et d'un comportement ; elle est identifiée par un identificateur unique.

La notion de relation issue de l'approche relationnelle permet de décrire un ensemble d'enregistrements. Le concept de relation est adaptée à des ensembles de données dont seule, la structure, est pertinente.

La représentation abstraite `MORE(BD)`, selon le modèle MORE, de la base de données analysée `BD` est définie par un ensemble `EClasses(BD)` d'entités-classes $\{C_1, \ldots, C_n\}$ et un ensemble `ERelations(BD)` d'entités-relations $\{R_1, \ldots, R_p\}$.

Le schéma abstrait $Sch_{MORE}(BD)$ décrivant `MORE(BD)` est défini par :

- la donnée du schéma `SchClasses` décrivant les entités-classes
- la donnée du schéma `SchRelations` décrivant les entités-relations
- la donnée de l'ensemble `Ass(BD)` des liens d'association
- la donnée de l'ensemble `Gen(BD)` des liens de généralisation-spécialisation

III.3.4. Analyse et Abstraction

Dans EXORE, la chronologie des tâches permet de séparer le processus d'identification des objets qui identifie également les anomalies et tente de les rectifier, du processus de construction des nouvelles structures qui exploite les résultats. La phase d'*analyse* du schéma existant permet :

- d'identifier les entités caractéristiques des relations sources
- de détecter les clés primaires candidates ainsi que les dépendances d'inclusion
- de détecter les dépendances fonctionnelles et normaliser les relations qui ne sont pas en troisième forme normale (3NF)
- de découvrir les similarités implicites entre attributs et résoudre les ambiguïtés dues au nommage incohérent

Cette phase résulte en un catalogue décrivant de manière détaillée les données existantes.
La phase d' *abstraction* ou de *construction du schéma cible* consiste à définir les différentes entités du schéma abstrait décrivant les données listées par le catalogue. La structure et les propriétés des relations sources servent à décrire les entités-relations et les entités-classes du schéma résultant. Ces deux phases sont schématisées dans la Figure III-1.

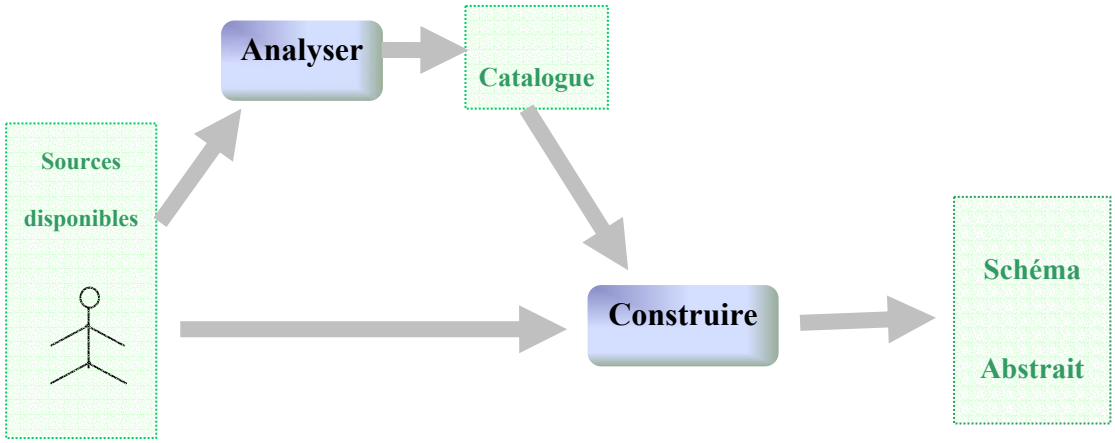


Figure III-1 Phases d'Analyse et de Construction

- **Phase d'analyse**

La phase d'analyse ne modifie pas la base de données source, mais produit un catalogue qui contient toutes les informations extraites et déduites de l'observation détaillée des structures de données et des informations disponibles. Le catalogue donne une description de la base constituée d'informations déduites et extraites à partir des diverses sources d'information ; il indique :

- pour chaque relation,
    - les attributs clés candidates détectées,
    - les attributs clés exogènes détectées ; une clé exogène représente un attribut qui permet d'identifier de manière unique un tuple d'une autre relation
    - les dépendances fonctionnelles déduites,
    - les attributs–collection détectés,
    - les régularités observées sur l'occurrence des valeurs nulles,
- et pour l'ensemble du schéma,
    - les dépendances d'inclusion,
    - les liens de similarités détectés entre attributs,
    - la donnée des relations ayant des clés primaires similaires,
    - la donnée des relations ayant même schéma, ou des schémas comparables.

Etude des similarités

L'analyse d'une base de données relationnelle commence, par l'identification des attributs-clés, dont la connaissance est essentielle pour déduire des liens entre différentes entités. Ainsi que nous l'avons vu précédemment, il semble peu fiable de fonder l'analyse de la base de données sur le nommage des attributs. Nous avons mis en évidence des solutions proposées [ANDE94, PETI96] consistant à exploiter certaines jointures. Bien que ces propositions constituent une avancée importante, leur intérêt pratique est limité par le fait qu'il n'existe pas toujours de jointure pour identifier des attributs synonymes. Nous définissons une nouvelle approche pour détecter des attributs qui se réfèrent au même domaine sémantique, mais ne sont pas définis avec le même nom, ou avec le même type effectif. Le principe mis en oeuvre se fonde sur la conjecture selon laquelle deux attributs de même sémantique sont utilisés « de la même manière » ou encore « dans le même contexte » ou des

contextes sémantiquement équivalents. Ainsi, notre stratégie consiste à définir et utiliser une similarité contextuelle. Le contexte d'utilisation d'un attribut se concrétise par l'ensemble des requêtes qui le manipulent. Nous supposons, ce qui est une hypothèse raisonnable, pouvoir stocker un volume (le plus important possible) de requêtes-utilisateurs pendant une certaine « période d'observation ». Nous étudions ensuite cette « base de requêtes » de manière à exploiter son volume et à extraire des connaissances sur le contexte d'utilisation des attributs. Les connaissances sont recherchées pour donner des indices sur la similarité contextuelle entre attributs. La technique mise en œuvre relève d'une approche de type « fouille de données » que nous présentons en détail à la section III.4.

Dépendances fonctionnelles

L'optimisation des requêtes selon des objectifs de performance conduit à réduire les jointures entre tables et ainsi à introduire des dépendances fonctionnelles à l'intérieur de plusieurs relations. Les algorithmes existants pour la recherche de dépendances fonctionnelles sont classés en deux catégories : les algorithmes orientés tuples et les algorithmes orientés attributs.

- *Les algorithmes orientés tuples* commencent par analyser les couples de tuples de la relation pour en extraire les dépendances fonctionnelles. Ces algorithmes utilisent souvent la notion *d'ensemble en accord* [MANN87, SAVN93]. Ces algorithmes sont coûteux en terme de temps d'exécution.
- *Les algorithmes orientés attributs* sont décomposés en deux phases : une phase de génération de candidats suivi d'une phase d'évaluation sur la base de données [KANT92, HUHT99]. Un des algorithmes les plus efficaces est connu sous le nom de TANE [HUHT99]. Ces algorithmes en majorité débutent en testant les dépendances fonctionnelles les plus simples (ayant un seul attribut en partie gauche) et poursuivent en générant et en testant des candidats de plus en plus complexes (ayant une partie gauche de plus grande taille). La différence entre ces algorithmes réside dans la stratégie d'évaluation des dépendances fonctionnelles candidates ainsi que dans l'élagage des candidats pour éviter des tests inutiles. TANE optimise le temps de recherche de ces dépendances fonctionnelles.

Dépendances d'inclusion

Les dépendances d'inclusion apportent un élément d'information important pour la déduction de liens sur des attributs contexte-similaires. Nous recherchons les dépendances d'inclusion sur des attributs qui ne sont pas nécessairement définis sur le même domaine mais sur des types-domaines comparables.

Clés candidates

Les clés candidates donnent des informations importantes ; en particulier, elles permettent d'identifier des clés étrangères. Une clé candidate est tout attribut dont le nombre de valeurs distinctes n'est autre que le nombre total de ses valeurs.

Attributs-collections

Il s'agit ici de détecter des attributs-collections implémentés par la création de plusieurs attributs de même type. Les attributs multivalués représentés à l'aide d'une relation multi-clés sont identifiés par des associations/agrégations

Etude des valeurs nulles

Nous étudions les régularités sur les occurrences de valeurs nulles qui indiquent la présence de tuples de « types » différents dans une même table. Nous cherchons à identifier des modèles ou motifs décrivant une relation du type de la relation `Article` (Tableau III-2) qui regroupe des tuples représentant des articles de types différents.

NumArt	Titre	Genre	AnAcq	Prix	Edit	Aut	Nbpag	Duree	Prod	Pays
000001	A	BD	1998	123	Eyrolles	Gardarin	150	NULL	NULL	NULL
000002	B	DM	1995	111	Dunod	Mannila	254	NULL	NULL	NULL
000003	C	Fiction	1972	150	NULL	NULL	NULL	1h30	W. B	USA
000004	D	Policier	1960	100	NULL	NULL	NULL	2h	MGM	France

Tableau III-2 La relation `Article` utilise des valeurs nulles

Le Tableau III-3 présente une relation `Article1` dans laquelle l'attribut `Cat` est un attribut discriminant indiquant la catégorie de l'article (`F` : Film ou `L` : Livre). Ainsi, si l'article est un livre, les attributs `Edit`, `Aut` et `Nbpag` sont renseignés alors que les attributs `Duree`, `Prod` et `Pays` ont une valeur nulle. De même, si l'article est un film, les attributs `Duree`, `Prod` et `Pays` sont renseignés alors que les attributs `Edit`, `Aut` et `Nbpag` ne le sont pas.

NumArt	Titre	Genre	AnAcq	Prix	Cat	Edit	Aut	Nbpag	Duree	Prod	Pays
000001	A	BD	1998	123	L	Eyrolles	Gardarin	150	NULL	NULL	NULL
000002	B	DM	1995	111	L	Dunod	Mannila	254	NULL	NULL	NULL
000003	C	Fiction	1972	150	F	NULL	NULL	NULL	1h30	W. B	USA
000004	D	Policier	1960	100	F	NULL	NULL	NULL	2h	MGM	France

Tableau III-3 La relation `Article1` un attribut discriminant et des valeurs nulles

Nous appliquons des techniques de fouille de données qui permettent de découvrir des classes de tuples ayant des structures identiques. Ceci est développé dans la section III.4.

Clés exogènes

Nous désignons par le terme de clé exogène, tout attribut qui joue le rôle de référence vers une autre entité, c'est à dire clé étrangère explicitement déclarée ou une référence implicite détectée lors de l'analyse. Nous recherchons ces clés en nous basant sur les similarités détectées. On identifie une clé exogène lorsque l'on trouve un attribut dans une relation qui est similaire à un attribut clé primaire ou clé candidate dans une autre relation, et si ces attributs ont également le même type ou sont liés par une dépendance d'inclusion qui a le même type et où on a une dépendances d'inclusion.

Relations similaires

L'objectif est de détecter des relations ayant des clés primaires, ou des clés candidates similaires. Nous supposons ici que l'étude du contexte d'utilisation et des autres critères ont permis d'identifier des attributs pour lesquels on a une forte probabilité de similarité. Pour identifier des clés similaires, nous imposons d'avoir des valeurs communes ou des types identiques. Dans cette étape on procède à la comparaison des schémas de relations pour identifier des schémas entièrement ou partiellement comparables ou des schémas sans attributs en commun.

- **Phase de construction**

La phase de construction utilise des heuristiques pour produire le schéma cible abstrait. Ainsi, certaines relations sources permettent d'identifier des entités-classes dont la structure est issue de la structure des relations sources. De même, l'exploitation de liens de similarités entre attributs permet d'identifier des clés étrangères ou des liens non explicitement exprimés. La notion de similarité permet de ne pas baser le processus d'analyse sur le nom des attributs, mais sur leur sémantique propre. Les règles de déduction utilisent également des dépendances d'inclusion entre attributs.
Les liens de généralisation/spécialisation peuvent être également détectés à partir d'une relation dans laquelle les occurrences de valeurs nulles suivent des règles ou motifs significatifs. Dans cet algorithme, sont détectées deux ensembles disjoints d'attributs qui prennent des valeurs nulles sur des lignes spécifiques de la relation. La détection de ces régularités donne lieu à la création d'une hiérarchie d'entités. La phase de construction consiste à construire un schéma de données selon le

modèle MORE défini plus haut, en exploitant les informations recueillies dans le catalogue et en interagissant avec l'expert.

Le schéma produit $Sch_{MORE}(BD)$ décrit un ensemble d'entités-classe $EC=\{EC_1, \ldots, EC_n\}$ et un ensemble d'entités-relation $ER = \{ER_1, \ldots, ER_p\}$. Il est entièrement déterminé par la donnée des schémas d'entités-classes, des schémas d'entités-relations, des associations et des liens d'héritage. Une étape initiale du processus consiste à utiliser l'ensemble DF des dépendances fonctionnelles détectées et à construire un nouveau schéma en 3NF. L'étude de la normalisation ne rentre pas dans le cadre de ce travail d'autant plus que des travaux antérieurs ont déjà étudié la normalisation des relations. Nous nous référons à des résultats existants [ABIT95, MANN92] pour appliquer un algorithme de normalisation qui transforme le schéma en 3NF.

Les tâches successives du processus d'abstraction sont les suivantes:

1. Identification des relations de base et des relations dépendantes parmi les relations en 3NF
2. Identification des structures optimisées et fusion
3. Création des entités de base
4. Identification des entités-association
5. Identification des liens associations
6. Identification des liens d'héritage
7. Identification des clés exogènes, attributs-collection et autres attributs
8. Création du schéma MORE complet avec identification des entités-classe et des entités-relation

A partir du schéma relationnel normalisé et de la donnée des clés exogènes, les relations de base et les relations dépendantes peuvent être identifiées ; nous appelons, ici, relation de base, une relation dont aucun attribut n'est une clé exogène et relation dépendante, toute relation qui n'est pas une relation de base. L'étape 2 permet de fusionner des relations résultant d'opérations d'optimisation ; cette étape utilise les schémas comparables listés dans le catalogue. A partir des relations de base et des schémas ainsi fusionnés, l'étape 3 peut identifier des entités-classe et des entités-relation. L'étape 4 utilise la liste des relations les entités-association et l'étape 5 identifie les liens d'association. L'étape suivante concerne les liens d'héritage qui sont identifiés en exploitant les informations du catalogue sur les relations clés-similaires, et les relations à schémas comparables ainsi que les régularités RegNull. L'étape 7 consiste à parcourir l'ensemble des entités détectées pour leur attribuer leurs clés exogènes et leurs attributs non clés pour construire le schéma final MORE. Les attributs collection sont alors

intégrés explicitement. L'étape 8 crée le schéma final en distinguant les entités-classes et les entités-relations.

- **Validité**

Dans les travaux sur le thème de la RCBD, le problème de la validité du processus défini est peu étudié. Cependant, R. Chiang et al. [CHIA94] listent plusieurs critères qui selon eux permettent de caractériser une « bonne » méthode.

Si nous faisons référence à ces travaux, la justesse, la complétude et la validité sont définies de la manière suivante :

La justesse indique le degré de fiabilité de la représentation de la base de données de départ dans le modèle final MORE.

Le principe de la rétro-conception est de produire une série de spécifications qui amènent à l'implémentation en question. Ainsi, selon cette approche un processus de RCBD est dit juste si, en partant du modèle abstrait final, et à travers de bonnes techniques théoriques de conversion, on aboutit à la base de données initiale.

La complétude est considérée satisfaite si toutes les structures implémentées dans la base de données sont prises en compte dans les règles de transition vers le modèle abstrait.

Selon la même approche, la validation consiste à vérifier que le modèle final est une bonne représentation du domaine de l'univers que la base de données doit représenter.

C. Ramanathan [RAMA97] fait également état de complétude et de consistance pour ce qui concerne le schéma obtenu par rétro-conception et évoque également l'efficacité du processus. La complétude est définie de la même façon, mais C. Ramanathan s'attache à démontrer que le résultat est complet tant au niveau des attributs que des données. La consistance est traduite par la préservation des dépendances fonctionnelles. Finalement, l'efficacité traduit le degré d'automatisation du processus de rétro-conception.

La validation, dans le sens où R. Chiang la définit nous semble difficile à établir par un processus automatique. Aussi, nous considérons la validité de la méthode selon deux critères : la préservation de l'information d'une part qui peut être vérifiée dans les algorithmes mis en œuvre et d'autre part l'adéquation du modèle à l'univers réel, dont la validation passe par un processus interactif.

- *La vérification de la préservation de l'information* concerne les attributs, les données et les dépendances fonctionnelles. L'étude des différents algorithmes mis en œuvre permet ce contrôle.

Dans cette étape, il suffit de parcourir les algorithmes de construction du schéma MORE et montrer la conservation des propriétés du schéma de départ. L'identification des entités revient à localiser ces relations dans le schéma source et les attributs correspondants et à les traduire directement dans le modèle MORE. Donc, il y a préservation des composants du schéma de départ.
En ce qui concerne l'identification des structures optimisées dans le cas d'un éclatement horizontal, la nouvelle entité créée a la même structure que celles de départ et l'ensemble des données est la réunion des tuples des deux relations. Dans le cas d'un éclatement vertical, il y a concaténation des structures et des données des tables pour reconstituer la table de départ. Dans cette étape les attributs et leur ensemble de valeurs sont invariants.
Les algorithmes de détection des hiérarchies et des sous-types, construisent des hiérarchies d'entités en réorganisant les structures. Aucun attribut n'est supprimé ; s'il n'est pas déclaré explicitement dans une entité, il est hérité.
- *La vérification de l'adéquation à l'univers réel*, fait nécessairement appel à l'intervention de l'utilisateur. Certains contrôles peuvent être partiellement automatisés mais sont dépendants du domaine spécifique de l'application. On note également que l'intervention de l'utilisateur peut être utile par exemple, pour donner un nom significatif à une association ou résoudre les conflits pouvant émerger lors de la détection d'une entité déjà en association avec une autre.

III.4. Techniques de fouille dans EXORE

Dans le contexte particulier de la RCBD, les processus courants d'analyse de la base de données existante, peuvent être déjà considérés comme des activités de fouille de données, dans la mesure où ils recherchent des dépendances, comparent des objets et ainsi scrutent les données et les structures existantes pour extraire des informations non explicites.
Dans la méthode EXORE, nous montrons que des techniques typiques de la fouille de données peuvent représenter un avantage pour la RCBD. Deux sources d'information sont principalement exploitées à cet effet : les données de la base et les requêtes utilisateurs. Nous utilisons les données pour extraire des régularités traduisant des liens sémantiques entre structures et les requêtes sont considérées comme source d'information implicite pour rechercher des similarités. L'exploitation de ces données très spécifiques, repose sur la conjecture selon laquelle l'utilisateur expert interroge la

base de données d'une manière qui reflète la sémantique des structures et des liens non explicites et que lui seul connaît.

Cette section présente deux types de fouilles réalisées dans la méthode EXORE afin d'extraire la sémantique sous-jacente**.** La première technique concerne la découverte de régularités dans les données d'une table et plus particulièrement à partir des attributs à valeurs nulles. La deuxième technique traite le problème de la détection de similarité entre attributs.

III.4.1. Régularités et valeurs nulles

Les liens de généralisation/spécialisation entre entités peuvent être implémentés de deux façons différentes dans une base de données relationnelle. D'une part, on peut utiliser les valeurs nulles de certains attributs dans une relation, d'autre part, on peut recourir à l'usage d'attributs clés primaires de même domaine avec des inclusions de valeurs.

Dans ce qui suit, nous nous intéressons à la détection de liens implémentés à l'aide de valeurs nulles. L'utilisation des valeurs nulles dans la structure d'une table relationnelle permet de détourner la structure d'une relation pour implémenter la notion de sous-type ; cette technique permet également d'optimiser les accès à des données a priori de types différents. Cet aspect des bases de données opérationnelles est peu étudié dans le cadre de la RCBD. Cependant, Wattiau et al. [WATT96] parlent des régularités de valeurs nulles dans une relation groupant deux ensembles d'attributs concernant des types distincts. Ainsi, pour une relation $R(\mathbf{CP}, K_1, ..., K_m, L_1, ..., L_n)$ où l'attribut ***CP*** clé primaire, cette méthode recherche à identifier une partition de la relation R en un ensemble de tuples où les attributs $K_1, ..., K_m$ ont des valeurs nulles et un ensemble de tuples où les attributs $L_1, ..., L_n$ ont des valeurs nulles. Les auteurs suggèrent alors de décomposer cette relation en deux relations ayant la même clé primaire et de remplacer la relation *R* par les deux relations $R_1(\mathbf{CP}, K_1, ..., K_m)$ et $R_2(\mathbf{CP}, L_1, ..., L_n)$.

Ramanathan [RAMA97] aborde cette question sous un autre aspect. Il suppose la présence d'un attribut discriminant dans la relation de départ et en déduit la création de trois relations avec des liens d'héritage. Ainsi, pour $R(\mathbf{CP}, K_1, ..., K_m, L_1, ..., L_n, Att_1, ..., Att_i, ..., Att_z)$ avec ***CP*** clé primaire, il suggère de rechercher un attribut discriminant Att_i pour lequel il existe une valeur $vali_1$ et un ensemble d'attributs $K_1, ..., K_m$ tels que si pour un tuple t, on a $Att_i = val_i$ alors les valeurs des attributs $K_1, ..., K_m$ dans t soient nulles. Il suggère alors l'éclatement vertical en trois classes

$C(\mathbf{CP}, Att_1, ..., Att_i, ..., Att_z)$, $C_1(\mathbf{CP}, K_1, ..., K_m)$ et $C_2(\mathbf{CP}, L_1, ..., L_n)$ où les classes *C1* et *C2* héritent de la classe *C*.

Dans EXORE nous supposons l'existence de ces deux cas de régularités dans les bases de données réelles et essayons de les résoudre automatiquement en utilisant des techniques de fouille de données.

Le cas des régularités de valeurs nulles avec un attribut discriminant Att_i est résolu à travers la technique des règles d'association de la forme suivante :

$$Att_i = val_1 \quad \rightarrow \quad K_1 = NULL\ ET\ \ldots\ ET\ K_m = NULL$$

avec des seuils de *support* et de *confiance* minimum fixés par le concepteur.

Dans le cas où la relation de départ ne contient pas un attribut discriminant, on peut recourir aux techniques de classification supervisée pour détecter les "sous-relations" éventuelles ainsi que leurs structures. Prenons, par exemple, la méthode basique des *K-moyennes* qui suppose que le nombre *K* de classes soit fixé. Cet algorithme débute par le choix aléatoire de *K* points dont chacun est considéré comme centroïde de sa classe. Puis, les autres points sont assignés à la classe du centroïde dont il est le plus proche. Le point moyen de chaque classe est ensuite calculé et considéré comme nouveau centroïde. Ces étapes se répètent jusqu'à ce que les classes soient stabilisées. Cette méthode s'adapte tout à fait à la détection des sous-relations qui nous intéressent.

En effet, la première étape consiste à détecter les relations candidates à l'éclatement en sélectionnant celles qui présentent une proportion importante de valeurs nulles. A chaque relation candidate, nous associons une matrice de valeurs binaires dans laquelle chaque ligne représente un tuple de la relation à partitionner : la valeur 1 correspond aux attributs ayant une valeur non nulle dans le tuple.

Ensuite, il s'agit de :

- sélectionner les attributs pertinents c'est à dire ceux qui prennent des valeurs nulles
- construire la matrice binaire à partir des attributs sélectionnés
- appliquer l'algorithme de classification sur les lignes de cette matrice

Pour mesurer la similarité entre tuples, nous avons retenu le coefficient de *Jaccard* qui permet de mesurer la distance entre deux tuples en terme de co-occurrences de valeurs sur les attributs des deux lignes. Pour deux lignes `i` et `j` de la matrice, si on note :

`q` le nombre d'attributs qui valent 1 sur `i` et sur `j`

`r` le nombre d'attributs qui valent 1 sur `i` et 0 sur `j`

`s` le nombre d'attributs qui valent 0 sur `i` et 1 sur `j`

Le coefficient de Jaccard qui définit la distance d(i,j) entre deux lignes est donnée par la formule $\frac{r+s}{q+r+s}$

Par exemple, soit la relation de départ donnée par la Tableau III-4,

Titre	Genre	AnAcq	Prix	Edit.	Aut.	Nbpag.	Duree	Prod.	Pays
A	BD	1998	12	Eyrolles	Gardarin	150	NULL	NULL	NULL
B	DM	1995	18	Mit	Manilla	254	NULL	NULL	NULL
C	Cours	1972	24	NULL	NULL	NULL	1h30	W.B	USA
D	Math	1980	20	Eyrolles	Coulomb	NULL	NULL	NULL	France

Tableau III-4 Relation avec régularités de valeurs

la classification représentée par la matrice binaire donnée dans le Tableau III-5.

Tuple	Edit.	Aut.	Nbpag.	Duree	Prod.	Pays
1	1	1	1	0	0	0
2	1	1	1	0	0	0
3	0	0	0	1	1	1
4	1	1	0	0	0	1

Tableau III-5 Matrice binaire associée à la relation

et dans ce cas, la classification produit la partition {{1, 2,4} , {3}}.

III.4.2. Recherche de similarités

La notion de similarité est un concept important dans le cadre des systèmes informatiques, comme dans d'autres sciences d'ailleurs. L'étude de la similarité entre objets, séquences, mots, documents … peut concerner le domaine de la linguistique, des mathématiques, des statistiques, ou de la biologie par exemple. La notion de similarité entre documents est une notion fondamentale pour de nombreuses applications en Traitement Automatique du Langage Naturel, et en particulier pour les applications destinées à exploiter l'information présente dans des bases de données documentaires de grande taille, telles que la recherche documentaire ou la structuration automatique de corpus (ensemble de discours oraux recueillis en vue d'une étude). En effet, la tâche de recherche documentaire consiste à chercher les documents les plus pertinents par rapport à un besoin d'information spécifié par une requête et peut donc être envisagé comme la recherche des documents les plus similaires à la requête. Les méthodes statistiques du traitement des langages naturels [DAGA99, LEE97] déterminent la probabilité d'une combinaison de mots à travers sa fréquence dans un échantillon de corpus. La manière la plus répandue de calculer la similarité entre documents est de représenter les documents dans un espace vectoriel et de considérer une mesure de similarité (au sens mathématique) entre les vecteurs représentant les documents.

Dans le cadre de l'analyse et la conception des systèmes d'information, la comparaison d'entités, de propriétés, de schémas, de bases de données est un thème central pour l'intégration de schémas, la retro-conception ou encore, la construction d'entrepôt de données. Le processus de conception d'une base de données consiste à définir des schémas de niveaux différents reflétant l'organisation des données. Ainsi, après analyse du domaine de l'application, on construit le schéma conceptuel puis le schéma logique. Le problème de synonymie intervient dès le début de ce processus, puisqu'il faut fixer une politique de nommage pour les attributs, les entités et certaines associations. Ces conventions de nommage aussi bien au niveau des attributs qu'au niveau des entités ont une influence sur le schéma logique de la base de données sur lequel le programmeur se base. D'ailleurs, la cohérence de nommage entre les différents composants (attributs, entités et associations) est un des critères de fiabilité du modèle conceptuel.

La question se pose également pour l'intégration de schémas qui répond à un double objectif : obtenir une représentation unifiée et non redondante de tous les composants d'un système, et prendre en compte l'évolution des schémas dans le temps. Ainsi, la difficulté majeure de l'intégration de schémas réside dans la détection et la résolution des conflits et des redondances qui peuvent exister entre plusieurs schémas. La notion de similarité peut aider à résoudre les conflits sémantiques (ou terminologiques) et les conflits structurels. En RCBD, le problème de similarité se pose lors de l'identification des attributs. Ainsi, tous les cas d'homonymie et de synonymie doivent être élucidés pour permettre ensuite, de détecter des similarités entre schémas.

En fouille de données, la recherche de régularités, motifs fréquents, ou classes d'objets nécessite également de pouvoir distinguer dans quelle mesure des données sont proches l'une de l'autre. Une mesure de similarité peut être définie a priori, mais un problème important réside dans l'évaluation de liens de similarité basée sur les données.

La notion de similarité dans le cadre des bases de données revêt différents aspects : on peut rechercher dans quelle mesure des objets stockés sous forme d'enregistrements sont proches ou bien étudier la similarité entre ensemble de données. De nombreux travaux ont contribué à définir des mesures de similarité entre objets, obtenues en questionnant les données. La similarité peut aussi bien concerner les structures syntaxiques que la sémantique attachée aux données. En fouille de données, l'analyse du "panier de la ménagère" consiste à recherche des similarités entre les objets représentant des clients et en basant le calcul sur la comparaison des valeurs représentant les produits qu'ils achètent. Pour ce qui est de la similarité sémantique, on peut citer V. Kashyap et A. Sheth [KASH96] qui définissent un cadre formel pour décrire la "proximité sémantique" entre objets de la base de données sans comparer

leurs structures respectives ; cette notion permet de comparer les objets du monde réel représentés dans la base à partir d'un "contexte de comparaison". L'approche s'applique de la même manière aux entités et aux attributs.

Dans certains cas, seule la similarité entre attributs est intéressante ; la recherche de similarités entre attributs peut servir à constituer des hiérarchies d'attributs ou des segments d'attributs qui, elles-mêmes permettent d'extraire des règles ou des modèles caractéristiques. Les liens de similarité entre attributs peuvent être fournies par des experts, elles peuvent être extraites d'un thésaurus. Mais, on ne dispose pas nécessairement de cette connaissance. Il est dans certains cas, important de disposer d'un moyen de mesurer la similarité. Les approches courantes de définition de mesure de similarité entre attributs reposent sur les valeurs de ces attributs. Différentes mesures doivent être définies pour prendre en compte différents types de similarité. Une approche traditionnelle dans la recherche de similarités consiste à évaluer une mesure basée sur les valeurs respectives de chaque attribut. Pour une application du type Panier de la ménagère cela permet d'identifier des produits achetés en général dans un même panier.

P. Moen [MOEN00] met en évidence l'insuffisance de ces mesures standard pour évaluer certains types de liens de similarité. Il présente, en particulier, un exemple du type "panier de la ménagère" dans lequel les attributs *lait, chips, moutarde, saucisses* sont à valeurs binaires ; la Figure III-2 donne le nombre d'enregistrements correspondant à chaque cas et des mesures standard de distance sont calculées. Elles s'avèrent insuffisantes pour différencier les trois cas. En effet, la valeur du χ^2 et les valeurs des distances d_1 et d_2 définies pour deux attributs binaires A et B par :

$$d_1 = \frac{fr(AB)}{fr(A) + fr(B) - fr(AB)}$$ où *fr()* représente la fréquence de l'attribut

$$d_2 = (1 - confiance(A \rightarrow B)) + (1 - confiance(B \rightarrow A))$$

donnent les résultats suivants : $\chi^2 = 0$ seulement dans le cas n°3 et permet uniquement de distinguer le cas d'indépendance. Les distances d_1 et d_2 valent 0 dans le cas n°2, ce qui conduirait à déduire que les produits *moutarde* et *saucisses* sont similaires et que *saucisses* et *chips* ainsi que *saucisses* et *lait* sont non similaires.

P. Moen introduit donc une mesure de distance externe permettant de mettre en évidence les différences entre attributs.

	saucisses=1	saucisses =0
chips=1	0	6
chips=0	6	0

Cas n°1 : association négative

	saucisses=1	saucisses =0
moutarde=1	6	0
moutarde=0	0	6

Cas n°2 : association positive

	saucisses=1	saucisses =0
lait=1	3	3
lait=0	3	3

Cas n°3 : indépendance

Figure III-2 Différents exemples de dépendances entre attributs

G. Das et H. Manilla [DAS98, DAS00] basent également la mesure de similarité entre deux attributs sur des facteurs externes aux attributs. Leur approche traite le cas d'attributs binaires ; elle est fondée sur l'utilisation de "preuves externes" apportées par d'autres attributs. L'idée centrale est de considérer deux attributs comme similaires, s'ils apparaissent (avec la valeur 1) non pas sur les mêmes lignes de la relation, mais avec les mêmes autres attributs ou avec des attributs similaires. [DAS98] introduit la notion de *similarité de contexte* qui prend en compte les autres attributs. Cette notion conduit les auteurs à considérer une distance prenant en compte les fréquences marginales d'un sous-ensemble d'attributs utilisé comme ensemble "preuve" ou référence de calcul.

Une première formule de distance externe entre attributs est définie comme suit :

$$d(A,B)= \sum_{Xi \in P} E(A,B,Xi)$$

où $P=\{X_1, \ldots, X_n\}$ est l'ensemble d'attributs "preuve"

$$E(A,B,X_i) = \left| fr(X_i,r_A) - fr(X_i,r_B) \right|$$

et $fr(X_i,r_A)$ représente la fréquence des lignes dans lesquelles l'attribut X_i prend la valeur 1 parmi les lignes de la relation dans lesquelles l'attribut A prend la valeur 1.

G. Das et H. Manilla proposent également d'utiliser des distances internes dans le calcul de distance externe définie par la formule $d(A,B)=|v_{A,P}-v_{B,P}|$ où $v_{A,P}=(d(A,X_1), \ldots, d(A,X_n))$.

L'algorithme décrit dans [DAS00] implémente un calcul de distance externe de ce dernier type. Cependant, le calcul est itératif et traduit ainsi une sorte de similarité croisée entre attributs et lignes de la relation. Deux attributs sont considérés similaires s'ils ont la valeur 1 sur des lignes non pas égales mais "similaires" ; la similarité est définie de manière circulaire puisque des lignes sont considérées comme similaires si elles utilisent des attributs similaires. Cet algorithme, présenté dans la Figure III-3, initialise les distances avec des valeurs aléatoires et les différentes exécutions ont démontré une convergence rapide.

```
{on suppose que les attributs sont rangés selon un ordre croissant}
debut
   Initialiser les distances avec des valeurs aléatoires
   Itérer
     Normaliser les distances de manière à ce que leur somme soit égale à 1
     Pour tout couple (A,B) d'attributs A,B de R avec A<B
       Extraire la sous-relation RA dont les lignes sont les lignes de R
                 dans lesquelles l'attribut A prend la valeur 1
       Extraire la sous-relation RB dont les lignes sont les lignes de R
                 dans lesquelles l'attribut B prend la valeur 1
       Calculer l'ensemble VA des vecteurs v obtenus
                 en appliquant la transformation f à toute ligne de RA
       Calculer l'ensemble VB des vecteurs v obtenus
                 en appliquant la transformation f à toute ligne de RB
       Calculer le vecteur moyenne MA des vecteurs de VA
       Calculer le vecteur moyenne MB des vecteurs de VB
       dtemp(A,B)← |VA-VB|
      finPour
     remplacer d par dTemp
   jusqu'à convergence
fin
```

Figure III-3 Algorithme de calcul de distance contextuelle proposé dans [DAS00]

Etant donné une ligne t = (t[A1], …, t[An]) de RA, la fonction f est définie de la manière suivante :

$$f(t)= (f(t[A1]), \ldots, f(t[A1]))$$

Pour tout attribut Ap, $f(t[Ap])= c(f_{A1}(t[Ap]), \ldots, f_{An}(t[Ap]))$

et $c(f_{A1},\ldots, f_{An}) = 1- \prod_{i=1}^{i=n}(1-f_{Ai})$ pour tout Ai tel que t[Ai]=1

$$f_{Ai}(t[Ap]) = \frac{K(d(Ai,Ap))}{\sum_{C\in R} K(d(Ai,C))} \text{ avec } K(X)=\frac{1}{1+X}$$

Ainsi, $f_{Ai}(Ap[t])$ peut être interprété comme la probabilité (basée sur la distance) pour que Ap soit proche de Ai et f(t[Ap]) comme la somme d'équivalence de Ap vis à vis des attributs utilisés dans t.

On peut voir `d(A,B)` comme une norme $\|v_A - v_B\|$ avec $v_A=(p_A(A1), \ldots, p_A(An))$ où $p_A(Ai)$ représente probabilité moyenne pour que Ai soit proche d'un attribut utilisé avec A et $v_B=(p_B(A1), \ldots, p_B(An))$.

Dans l'approche de rétro-conception EXORE, la notion de similarité est liée au domaine sémantique ; deux attributs doivent être considérés comme similaires s'ils sont définis sur le même domaine sémantique. La phase de reconstruction peut ainsi identifier des attributs représentant la même propriété du monde réel et leur associer le même domaine sémantique. Ceci concerne les clés primaires, clés étrangères, mais également les attributs non-clés qui ont des synonymes. Selon la théorie des bases de données relationnelles, le domaine sémantique définit le rôle exact de l'attribut : un attribut devrait être reconnu comme clé étrangère par l'identification de son domaine sémantique uniquement. Cependant, dans les SGBD, les types sur lesquels sont définis les attributs sont trop généraux et insuffisants pour identifier des domaines identiques. Nous avons discuté des autres critères permettant d'identifier deux attributs identiques : le nom, l'ensemble des valeurs effectives dans la base, l'occurrence d'opérateurs comme certaines jointures dans les requêtes. L'occurrence de jointures et les valeurs communes des ensembles de valeurs sont des critères intéressants comme le montrent les exemples vus au chapitre II, mais ne sont pas toujours disponibles. Aussi, nous présentons, dans cette section, un approche basée sur des critères externes ou critères d'utilisation des attributs. L'idée centrale est inspirée des recherches présentées ci-dessus sur les distances contextuelles entre attributs. Dans le cadre de la retro-conception où l'on dispose d'une base de données ancienne, le contexte d'utilisation de la base peut apporter des informations pertinentes. Dans EXORE, la notion de *similarité contextuelle* entre attributs se fonde sur la mise en place d'un "réservoir" de requêtes, ou base de requêtes (BQ) obtenues après une période d'observations de la base de données en usage. La base de requêtes représente le *contexte d'utilisation* des attributs. Certaines bases sont en permanence accédées et questionnées à partir de terminaux et sites web ; on peut disposer ainsi rapidement d'une masse importante de transactions contenant des requêtes exprimées sur la (ou les) bases accessibles.

La notion de similarité contextuelle utilise donc le contexte d'utilisation des attributs dans les requêtes. Ainsi, nous évaluons la similarité de deux attributs en mesurant la distance entre leurs contextes d'utilisation et en utilisant également un autre critère comme la comparaison des types déclarés dans la base de données source ou l'ensemble des valeurs prises. Nous donnons également une interprétation circulaire de la similarité en posant deux attributs comme similaires s'ils sont utilisés dans des requêtes où interviennent les mêmes attributs ou bien des attributs similaires. Cette notion est traduite par une définition récurrente. Dans la suite, nous donnons quelques définitions et notations plus formelles de la distance contextuelle pour EXORE.

On suppose les requêtes exprimées en langage SQL et que toute requête est de la forme :

```
select      <liste-attributs>
from  <liste-relations>
where <condition>;
```

La base de requêtes ***BQ*** est un ensemble dont les éléments représentent les différentes requêtes SQL soumises à la base de données pendant une période d'observation.

Contexte d'utilisation

On définit le *contexte d'utilisation* d'un attribut *R.A* d'une relation R et on note *ContexteU(R.A)*, l'ensemble des requêtes de *BQ* dans lesquelles *R.A* est présent.

Nous utilisons particulièrement les opérations de jointure et de sélection qui apparaissent dans les requêtes. Soit ***A***=*Att(BD)* l'ensemble des attributs de la base de données *BD*

Une expression de jointure ou *J-expression* sur les attributs *X* et *Y* de ***A*** est notée

$$X\ op\ Y \text{ avec } op \in \{=, <, >\}$$

Une expression de condition ou *C-expression* sur un attribut *X* de ***A*** est notée

$$X\ op\ val \text{ avec } op \in \{=, <, >\} \text{ et } val \in \{X\}$$

Une expression de sélection ou *S-expression* sur un attribut X de ***A*** est notée

$$SELECT\ X \textbf{ avec } X \in \{A\}$$

Pour une requête *q* de *BQ*, on note *W(q)* l'ensemble des *J-expressions*, *C-expressions* et *S-expressions* apparaissant dans *q*.

Pout tout attribut A_i de ***A***, $W(q)[A_i]=1$ est vrai si *Ai* est un attribut d'une *J-expression* ou d'une *C-expresion* de *W(q)*. Dans ce cas, on dit que *Ai* est *présent* dans *W(q)* sinon $W(q)[A_i]=0$.

Pour une requête *q* de *BQ*, on note $Att(q))=\{X \in A\ ;\ W(q)[A_i]=1\}$

Pour un ensemble Q de requêtes de BQ, on note $Att(Q)= \bigcup_{q \in Q} Att(q)$

Donc, $Att(Q)=\{X \in A \ ; \ \exists q \in Q \ / \ W(q)[A_i]=1\}$

Pour tout X de $\boldsymbol{A}$, on a défini le contexte d'utilisation de X, comme l'ensemble des requêtes de BQ dans lesquelles X est présent, donc :

$$ContexteU(X) = \{q \in BQ \ ; \ W(q)[X] = 1 \}$$

Si un attribut Ai appartient à $Att(ContexteU(X))$, on dit que $ContexteU(X)$ **utilise** Ai.

La similarité entre deux attributs est évaluée selon leur contexte d'utilisation. On définit donc la similarité de deux contextes d'utilisation. Deux contextes d'utilisation sont considérés comme similaires s'ils utilisent les mêmes attributs ou si chacun des deux utilise des attributs similaires à ceux utilisés par l'autre.

Contextes d'utilisation similaires

On dit que les contextes d'utilisation des attributs X et Y sont *similaires* et on note $ContexteU(X)$ $\underline{sim}$ $ContexteU(Y)$ si on a :

soit $Att(ContexteU(X)) = Att(ContexteU(Y))$ soit $Att(ContexteU(X)) / Att(ContexteU(Y))$ et $Att(\boldsymbol{ContexteU}(Y)) / Att(\boldsymbol{ContexteU}(X))$ sont *contexte-similaires*

Attributs contexte-similaires

Deux attributs X et Y sont dits *contexte-similaires* si leurs contextes d'utilisation sont similaires. On note $X \ sim_C \ Y$. On peut remarquer que la relation de similarité contextuelle sim_C définit une relation d'équivalence sur l'ensemble des attributs. Deux attributs X et Y sont dits *directement contexte-similaires* si $Att(ContexteU(X)) = Att(ContexteU(Y))$ et

$\{X\} \cap \{Y\} = \varnothing$ ou $type(X)$ et $type(Y)$ compatibles

Ensemble d'attributs contexte-similaires

Soient $A1$ et $A2$ deux ensembles d'attributs inclus dans $\boldsymbol{A}$, $A1$ et $A2$ sont dits *contexte-similaires* si

$\forall X \in A1 \ , \ \exists Y \in A2$ tel que X et Y soient *contexte-similaires*

et $\forall Y \in A2 \ , \ \exists X \in A1$ tel que X et Y soient *contexte-similaires*

Notation : on note $Att_{CU}(X)$ l'ensemble $Att(ContextU(X))$

Distance contextuelle

Le processus de calcul de la distance contextuelle opère sur une matrice représentant la base de requêtes comme le montre le Tableau III-4, dans laquelle chaque ligne représente une requête. Les colonnes de la matrice représentent les attributs et tout élément (i,j) est égale à $W(q_i)[X_j]$.

	X1	**X2**	**X3**	**…**	**Xm**
Q1	0	0	1		1
Q2	0	1	0		1
…					
Qi	1	0	1		1
…					
Qn	1	1	1		0

Tableau III-4 Représentation d'une base de requêtes BQ

On définit la distance contextuelle entre deux attributs relativement à la base de requêtes BQ. On peut remarquer que la similarité contextuelle de EXORE est définie de manière circulaire et dans les mêmes termes que la similarité contextuelle définie dans [DAS98]. Le contexte d'un attribut est ici assimilé à la matrice binaire représentant les requêtes utilisant l'attribut. Le calcul de distance est ainsi ramené à un calcul de distance contextuelle entre attributs binaires. Nous utilisons une approche semblable à celle définie par Das et Manilla. La distance contextuelle d(A,B) entre deux attributs A et B est donc évaluée comme la norme $|v_A - v_B|$ où $v_{A,P}$ =(v_A=(p_A (A1), …, p_A (An))) où p_A (Ai) peut être interprétée comme la probabilité pour qu'un attribut Ai soit proche d'un attribut du contexte d'utilisation de A. La Figure III-4 schématise ce calcul.

Dans l'exemple 1 donné par le Tableau III-5, on suppose avoir cinq attributs `A, B, C, D` et `E` avec cinq requêtes.

	A	**B**	**C**	**D**	**E**
Q1	1	0	1	1	1
Q2	0	1	1	1	1
Q3	0	0	1	0	0
Q4	0	0	0	1	0
Q5	0	0	0	0	1

Tableau III-5 Exemple 1

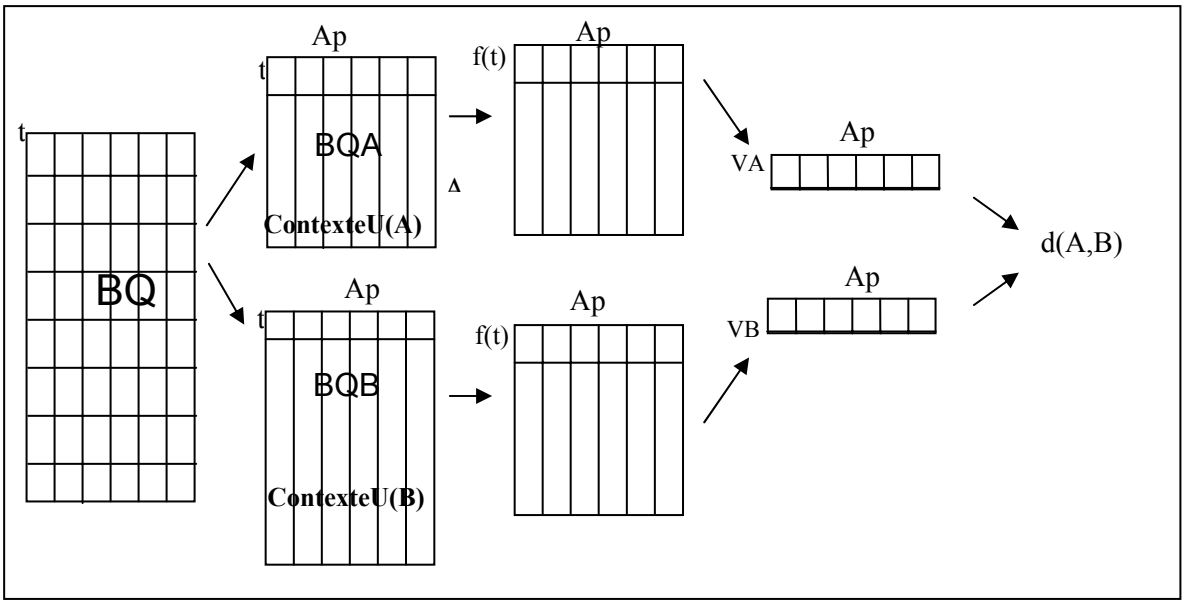


Figure III-4 Calcul de distance contextuelle selon BQ

En analysant cette matrice on constate que :

$ATT_{CU}(A) = \{C, D, E\}$, $ATT_{CU}(B) = \{C, D, E\}$, $ATT_{CU}(C) = \{A, B, D, E\}$ et $ATT_{CU}(D) = \{A, B, C, E\}$ $ATT_{CU}(E) = \{A, B, C, D\}$

Les valeurs obtenues après 10 itérations donnent les distances suivantes :

d(A,B) = 0 d(D,C) = 0 d(E,C) = 0 d(E,D) = 0 d(C,B) = 0.0172
d(D,B) = 0.0172 d(E,B) = 0.0172 d(C,A) = 0.0172 d(D,A) = 0.0172
d(E,A) = 0.0172

On obtient ainsi une indication de similarité contextuelle entre les attributs A et B d'une part et entre les attributs C, D, E d'une autre part. On peut justifier ceci en observant que les attributs A et B sont utilisés avec les mêmes attributs C, D et E et dans les mêmes proportions. De même les attributs C, D et E sont utilisés avec les attributs A et B. A et B étant proches selon cette distance de similarité, il est donc naturel que C, D et E soient évalués comme proches .

Dans l'exemple donné ci-dessous, nous considérons les attributs du schéma suivant :

Abonne(IdAbonne, DateDebutAbon, DateFrinAbon)
Emprunt(NumEmp, IdPersonne, DateDeb)

Supposons avoir relevé les requêtes Q1 à Q6 données ci-dessous :

```
Q1 : SELECT NumEmp FROM Emprunt, Abonne
     WHERE Abonne.IdAbonne = Emprunt.IdPErsonne;
Q2 : SELECT DateDebAbon FROM Emprunt, Abonne
     WHERE  Abonne.IdAbonne = Emprunt.IdPErsonne;
Q3 : SELECT Emprunt.DateDeb FROM Abonne, Emprunt
     WHERE Emprunt.DateDeb > Abonne.DateDebAbon;
Q4 : SELECT Count(*) FROM Abonne, Emprunt
     WHERE  Emprunt.DateDeb > Abonne.DateFinAbon
Q5 : SELECT DateFinAbon from Abonne;
Q6 : SELECT DateDeb from Emprunt;
```

La matrice binaire associée est donnée par le Tableau III-8..

	A NumEmp	B DateDebAbon	C DateFinAbon	D DateDeb	E IdAbonne	F IdPersonne
Q1	1	0	0	0	1	1
Q2	0	1	0	0	1	1
Q3	0	1	0	1	0	0
Q4	0	0	1	1	0	0
Q5	0	0	1	0	0	0
Q6	0	0	0	1	0	0

Tableau III-6 Exemple 2

En analysant cette matrice, on constate que

$Att_{CU}(A) = \{E, F\}$ $Att_{CU}(B) = \{D, E, F\}$ $Att_{CU}(C) = \{D\}$

$Att_{CU}(D) = \{B, C)$ $Att_{CU}(E) = \{A, B, F\}$ $Att_{CU}(F) = \{A, B, E\}$

Le calcul de distance converge rapidement et donne les distances suivantes :

$d(F, E) = 0.0$ $d(E, A) = 0.00005$ $d(F, A) = 0.00005$

$d(D, C) = 0.0019$ $d(E, B) = 0.011$ $d(F, B) = 0.011$

Le calcul de distance contextuelle permet donc d'identifier une synonymie possible entre les attributs F et E. D'après les résultats, on peut également poser l'éventualité d'une synonymie entre E et A ou entre F et A par exemple.
L'étude des informations additionnelles est nécessaire pour déduire un lien de similarité contextuelle ; aussi le type des attributs et de leurs ensembles de valeurs respectifs sont examinés.
Ceci permet de conclure à une forte probabilité pour un lien de similarité entre `F(IdPersonne)` et `E(IdAbonne)` s'ils ont un nombre important de valeurs communes dans la base de données source.

III.5. Conclusion

Nous avons défini dans ce travail, qui a fait l'objet d'une thèse [BARB00], une nouvelle approche pour la rétro-conception de bases de données relationnelles. La méthode EXORE exploite le volume souvent très important des informations disponibles, par des techniques de fouilles de données.
La méthode EXORE offre une approche réaliste, sans a-priori accommodant sur l'état des données. Elle prend en compte la présence d'anomalies et les incohérences inhérentes au fonctionnement d'une base de données en exploitation. En effet, dans le cas des bases de données réelles, utilisées dans l'industrie, on observe de nombreuses déviations par rapport aux modèles décrits dans le cadre académique. La théorie des bases de données relationnelles définit des normes sur les structures et les données, qui ne sont pas nécessairement respectées dans les bases de données industrielles. Les déviations par rapport au modèle théorique sont dues à des erreurs de conception et d'implémentation, à une sous-estimation de l'importance de ces normes pour la cohérence du système et, enfin, à des impératifs de performance dans l'exécution des transactions.
EXORE se focalise particulièrement sur les anomalies de nommage et les modifications apportées sur les bases de données pour accroître les performances des systèmes. La première phase de la méthode détecte donc les cas de nommage suspects en recherchant les attributs synonymes ; elle détecte également les relations ayant des schémas proches et susceptibles d'être fusionnés.

La plupart des travaux antérieurs dans le domaine de la RCBD proposent un processus de rétro-conception aboutissant à un schéma conceptuel Entité-Association (EA) ou Entité-Association Etendu (EAE). Nous avons défini pour EXORE, un modèle de données intermédiaire appelé MORE (Modèle Objet pour la REtro-conception) pour exprimer le schéma cible. Les bases de données actuelles

susceptibles de nécessiter une reconstruction, sont de type relationnel. La simplicité des structures du modèle relationnel et leur fondement mathématique contribuent à leur efficacité et à leur robustesse. Cependant, le modèle objet offre des possibilités de modélisation beaucoup plus étendues. Aussi, nous avons défini un modèle intégrant les notions de « relation » et de « classe » facilitant une implémentation dans un SGBD objet-relationnel.

Un processus EXORE est décomposé en deux phases distinctes d'analyse du schéma existant et de construction du schéma cible. Cette dichotomie permet de séparer le processus d'identification qui détecte les anomalies du processus de construction qui produit le schéma cible.

Chapitre IV Bilan et Perspectives

Nous avons présenté deux contributions dans le domaine de la fouille de données d'ordre méthodologique et applicatif. Dans ce chapitre, nous établissons un bilan et nous présentons les perspectives ouvertes par ces travaux ainsi que deux nouveaux axes de recherche que nous avons récemment initiés. Nous montrons également comment ces activités de recherche s'inscrivent dans la perspective du nouveau projet **EXeCO** au laboratoire I3S.

Une méthodologie basée sur l'optimisation

Concernant le problème de la découverte des motifs, nous avons proposé une démarche méthodologique basée sur un processus d'optimisation. Ceci nous a conduit à étudier la qualité d'une règle et les différents critères qui peuvent la caractériser.

Nous avons étudié les caractéristiques des espaces de recherche et déterminer en quoi certains algorithmes sont plus adaptés. Les propriétés des espaces de recherche (combinatoire élevée, existence de minima locaux, indices antagonistes) justifient le recours à des algorithmes évolutionnaires. Nous avons effectivement montré que, dans certains cas, ces algorithmes permettaient de découvrir des motifs ignorés par d'autres approches. Aussi, le bilan de cette étude est essentiellement d'ordre méthodologique ; il suggère que la recherche des meilleures règles débute en identifiant clairement le sens donné à la qualité, l'intérêt ou l'utilité des règles et les indices choisis pour évaluer ces critères, puis en étudiant l'espace de recherche produits par ces indices.

Notre étude s'est focalisée sur le caractère objectif de l'intérêt. Cependant, il est essentiel pour la pertinence de la démarche, d'étudier le caractère subjectif de l'intérêt des modèles en termes d'utilité pour les experts du domaine. Ce sujet soulève des difficultés car les facteurs de qualité dépendent alors du domaine des données et visent à sélectionner des informations intéressantes pour un utilisateur donné ou innovantes en regard de connaissances déjà acquises sur le domaine. Ce travail trouvera un

terrain d'expérimentation adéquat dans le cadre du contrat de recherche avec la CNAF (Caisse Nationale d'Allocations Familiales) pour extraire des modèles, règles ou motifs séquentiels traduisant le comportement des allocataires ainsi que leur relation avec leurs interlocuteurs à l'intérieur des centres. En effet, la connaissance acquise sur les allocataires par les experts métier des CAF fournira un contexte adapté pour valider des solutions relatives à l'intérêt subjectif des modèles et à leur effet de surprise. Cette recherche sera réalisée dans le cadre d'une thèse financée par la CNAF.

Application de la fouille de données à l'ingénierie des systèmes d'information

Le champ d'application de la fouille de données est vaste ; tout système ou toute activité générant des données en grand nombre est potentiellement un domaine de prédilection. Malgré cela, l'application de techniques de fouille de données dans le cadre de l'ingénierie des systèmes d'information (SI) n'est pas courante.

Nous avons proposé la méthode de retro-conception EXORE qui exploite par des techniques de fouilles de données, le volume souvent très important des informations disponibles dans une base de données. Cette méthode offre une approche réaliste et prend en compte la présence d'anomalies et d'incohérences inhérentes au fonctionnement d'une base de données en exploitation. EXORE se focalise particulièrement sur les anomalies de nommage et les modifications apportées sur les bases de données pour accroître les performances des systèmes. La méthode détecte les cas de nommage suspects en recherchant les attributs synonymes ainsi que les relations ayant des schémas proches et susceptibles d'être fusionnés.

L'expérience initiée avec la méthode EXORE doit être enrichie par la mise en œuvre d'autres techniques. Le modèle EXORE considère des bases de données en usage éventuellement dégradées. A ce sujet, nous avons proposé des solutions aux problèmes des nommages incohérents, de l'usage de valeurs nulles, de la détection de leurs sémantiques cachées ou de structures optimisées. Il serait également intéressant d'analyser l'usage des valeurs par défaut qui peuvent permettre d'extraire des connaissances insoupçonnées.

Nous avons distingué les notions d'entité-classe et d'entité-relation dans le modèle sous-jacent MORE de manière à pouvoir étendre le processus au cadre plus général de la retro-conception d'applications. C'est dans ce contexte que pourront être détectées les différentes opérations caractéristiques d'un type de données. Cet axe de recherche méritera d'être développé et nécessitera de combiner l'étude du code de l'application et des requêtes SQL imbriquées avec l'analyse des données.

Enfin, pour envisager une implémentation future du schéma MORE extrait, il sera important de poursuivre l'étude en décrivant la phase de traduction vers un modèle logique donné, par exemple le modèle relationnel-objet.

Les travaux de recherche que nous avons mené sur l'application des techniques de fouille de données pour la ré-ingénierie des systèmes d'information se sont inscrits dans le cadre du projet MECOSI (MEthodes de Conception des Systèmes d'Information). Ils constituent une première étape dans les projets que nous nous attachons à développer dans le nouveau projet EX**e**CO (**EX**traction de connaissances à partir des données **e**t analyse et **CO**nception des systèmes d'information) qui succède au projet MECOSI et qui a pour objectifs de développer non seulement les deux axes *Extraction* et *Systèmes d'information*, mais également les sujets qui les fédèrent comme la réingénierie des SI.

Application de la fouille de données en bioinformatique

En ce qui concerne l'axe *Fouille de données*, nous souhaitons développer le champ des applications. Parmi les domaines dans lesquelles la fouille de données peut apporter des solutions, la technologie des puces à ADN paraît un des plus prometteurs. Un projet a démarré sur ce thème, en juillet 2003 ; il fait l'objet d'une thèse.

Une biopuce, ou puce à ADN, permet aujourd'hui de mesurer l'expression de plus de 40 000 gènes dans une cellule ou un tissu et de comparer l'expression de ces gènes entre différentes conditions (normal/pathologique, traité/non traité, cinétiques temporelles, …). L'analyse statistique des données obtenues consiste dans un premier temps à sélectionner parmi l'ensemble des gènes exprimés un sous-ensemble de gènes qui ont été significativement modulés puis dans un second temps à regrouper ces gènes en des sous-ensembles (clusters) présentant une homogénéité de profils d'expression par rapport aux différentes conditions expérimentales. Des méthodes de clustering hiérarchique sont souvent mises en œuvre à cet effet.

L'intérêt du clustering des données d'expression est de mettre en évidence des sous ensembles de gènes qui bougent de façon coordonnée dans le temps ou dans l'espace et qui peuvent de ce fait faire partie de réseaux complexes de régulation génique. De tels réseaux sont connus pour être impliqués dans certains processus physiologiques ou physiopathologiques et prennent toute leur importance quand ils incluent des éléments de contrôle clés comme des enzymes ou des récepteurs qui sont des cibles privilégiées pour une modulation par des molécules pharmacologiques. En particulier, pour l'industrie pharmaceutique, une question clé dans l'exploitation des données d'expression génomiques

est : existe-t-il une enzyme ou un récepteur caché derrière un cluster qui pourrait être utilisé pour traiter une pathologie ou un disfonctionnement physiologique ?

Aujourd'hui, les programmes de génomique génèrent des masses de données répertoriées dans des bases de données internationales accessibles en ligne. Ces données se présentent soit sous forme d'informations brutes sur les séquences ou sur des résultats d'expérience (data) soit sous forme d'informations bibliographiques et d'annotations (texte) représentant les connaissances acquises sur les gènes ou les protéines et leurs interactions. Un enjeu majeur de la bioinformatique est d'extraire de ces textes des éléments pertinents aidant à la classification des gènes.

L'objectif de ce projet consiste à rechercher si l'analyse des sources bibliographiques associées à chaque gène d'un cluster permet de donner une cohérence et une unité au cluster en montrant l'existence d'un nombre limité d'enzymes ou de récepteurs qui lui est spécifiquement associé. Pour cela, un ensemble de gènes modulés par un produit seront étudiés. Pour chaque gène on disposera de données d'expressions mesurées et d'un ensemble d'annotations "texte" sélectionnées par les biologistes.

Une première étude consistera à analyser les annotations de manière à extraire pour chaque gène les noms d'enzymes et de récepteurs cités dans les textes. Pour les récepteurs, le processus devra utiliser des thesaurus de termes synonymes. On obtiendra ainsi une description de chaque gène sous forme d'un profil bibliographique "enzymes récepteurs cités dans la bibliographie du gène". Ce travail nécessitera de définir une structure de données adéquate pour représenter et stocker un profil. Il imposera ensuite le choix circonstancié d'algorithmes de fouille de texte adaptés aux objectifs décrits.

Une seconde étude concernera à mettre en œuvre des algorithmes de classification non supervisée (clustering) sur les données d'expression pour obtenir des classes homogènes de gènes. On testera différents critères de voisinage et différents types d'algorithme pour obtenir des frontières robustes de clustering même en présence de bruit.

La dernière étape consistera à construire un classifieur permettant de prédire le cluster auquel appartient un gène à partir de son "profil bibliographique". On appliquera donc des méthodes d'apprentissage supervisée à ces données de profil étiquettées par le nom d'un cluster. Il s'agira tout d'abord de sélectionner les propriétés les plus discriminantes d'un profil pour déterminer le classement d'un gène. On devra également, évaluer différentes techniques d'apprentissage pour extraire les modèles de classifieur les meilleurs. Dans ce contexte, la qualité d'un modèle sera évaluée par sa précision prédictive (taux d'exemples bien classés) et par sa lisibilité (nombre de propriétés utilisées en entrée). On confrontera les résultats obtenus de manière à vérifier, à l'aide de connaissances extraites

de la littérature sur les interactions entre gènes, enzymes et récepteurs, si les enzymes ou récepteurs associés spécifiquement à chaque cluster ont une pertinence biologique. Des expérimentations devront être menées pour permettre de tirer des conclusions sur la pertinence de cette approche qui rentre dans le domaine en plein essor actuellement du text mining et qui vise à l'extraction de connaissances inédites à partir de vastes collections de données se présentant sous forme de textes.

Références

[ABIT95] S. Abiteboul, R. Hull and V. Vianu. Foundations of Databases. Addison-Wesley Publishing Company,1995.

[AGRA93] R. Agrawal, T. Imielinski, A. Swani. Mining Association Rules between sets of items in large databasess. Proc. Int.Conf. on Management of Data, SIGMOD, 1993.

[AGRA94] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. Proc.of the 20th international conference on Very Large Data Bases (VLDB'94), p. 428 – 499. Morgan Kaufmann, 1994.

[AIKE99] P. Aiken, O. K. Ngwenyama and L. Broome. Reverse-Engineering New Systems for Smooth Implementation. IEEE Software, 1999.

[AKOK98] J. Akoka, I. Wattiau. MeRCI : An Expert System for Software Reverse Engineering. Fourth World Congress on Expert Systems, p. 209-217, Mexico, 1998.

[ALI92] K. M. Ali et M. J. Pazzani. Reducing the Small Disjuncts Problem by Learning Probabilistic Concept Descriptions. In T. Petsche editor, Computational Learning Theory and Natural Learning Systems, Volume 3, 1992.

[ANDE94] M. Andersson. Extracting an Entity Relationship Schema from a Relational Database through Reverse Engineering. Proc. of the 13th Conf. on ER Approach, Manchester, UK, 1994.

[ANKE98] M. Ankerst, B. Braunmüller, H. P. Kriegel and T. Seidl. Improving Adaptable Similarity Query Processing by Using Approximations. In A. Gupta, O. Shmueli and J. Widom Ed., Proc. of the 24th Int. Conf. on Very Large Data Bases (VLDB'98), p.206-217, New York, USA, 1998.

[ARAU00] D.L.A. Araujo, H.S. Lopes et A.A. Freitas. Rule discovery with a parallel genetic algorithm. Genetic and Evolutionary Computation (GECCO-2000) Workshop, p. 89-92. Las Vegas, USA. 2000.

[ARAU99] D.L.A. Araujo, H.S. Lopes, A.A. Freitas. A parallel genetic algorithm for rule discovery in large databases. Proc. of the 1999 IEEE Systems, Man and Cybernetics Conf., p.940-945, Tokyo, 1999.

[ATKI89] M.Atkinson, F.Bancilhon, D.DeWitt, K.Dittrich, D.Maier, S.Zdonik. The Object-Oriented Database System Manifesto. First Int. Conf. on Deductive and Object-Oriented Databases, 1989.

[AUGI95] S.Augier, G.Venturini et Y.Kodratoff. Learning first order logic rules with a genetic algorithm. Proc. of the First Int. Conf. on Knowledge Discovery and Data mining, Montréal, 1995.

[BARB00] A. Barbar. Fouille de données et de texte – Application à la rétro-conception de systèmes d'informations. Poster, Forum Jeunes Chercheurs, INFORSID, Lyon, 2000.

[BARB01a] A. Barbar, M. Collard. Attribute similarity : a data mining issue. Advances in Intelligent Data Analysis (AIDA'01), Int. ICSC Congress on Computational Intelligence, p. 215-221, Bangor, Wales, 2001.

[BARB01b] A. Barbar. A User Driven Method for Database Reverse Engineering. Conf. on Advanced Information Systems Engineering, CAiSE'01, The 8th Doctoral Consortium, Interlaken Switzerland, 2001.

[BARB02] A. Barbar, Extraction de connaissances pour la retro-conception d'une base de données vers un schéma objet, Thèse de l'Université de Nice-Sophia Antipolis, 2002.

[BATI92] C. Batini, S. Ceri and S. Navathe. Conceptual Database Design : an Entity-Relationship Approach. Benjamin Cummings, 1992.

[BAYA98] R. J. Bayardo. Efficiently mining long patters from databases. In proceedings of the 1998 ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'98), p. 85-93, ACM Press, 1998.

[BAYA99] R. J. Bayardo and R. Agrawal. Mining the most interesting rules. Proc. of the 1999 Int. Conf. on Knowledge Discovery and Data Mining (KDD'99), p. 145-154, San Diego, CA, 1999.

[BEHM97] A. Behm, A. Geppert, and K. R. Dittrich. On migration of relational schemas and data to object-oriented database systems. Proc. of the 5th Int. Conf. on Re-Technologies in Information Systems, Klagenfurt, Austria, 1997.

[BENN99] K. P. Bennett, U. Fayyad and D. Geiger. Density-based Indexing for Approximate Nearest-neighbor Queries. Proc. of the 5th Int. Conf. on Knowledge Discovery and Data Mining (KDD'99), p. 233-243, San Diego, CA, 1999.

[BESAN01] R. Besançon, J. C. Chappelier M. Rajman, and A. Rozenknop. Improving text representations through probabilistic integration of synonymy relations. Proc. of the Tenth Int. Symposium on Applied Stochastic Models and Data Analysis (ASMDA'2001), p. 200-205, Compiègne, France.

[BHAT99] S. Bhattacharyya. Direct Marketing Performance Modeling using Genetic Algorithms. INFORMS Journal of Computing, 11(3), 1999

[BHAT00] S. Bhattacharyya. Evolutionary Algorithms in Data Mining : Multi-Objective Performance Modeling for Direct Marketing. Proc. of the Sixth Int. Conf. on Knowledge Discovery and Data Mining (KDD-2000), Boston, AAAI Press, 2000.

[BLAH95] M. Blaha. Observed Idiosyncracies of Relational Database Designs. 2nd Working Conf. on Reverse Engineering, Toronto, Ontario, 1995.

[BLAH98a] M. Blaha. On Reverse Engineering of Vendor Databases. 5th Working Conf. on Reverse Engineering, Honolulu, Hawaii, 1998.

[BLAH98b] M. Blaha and W. Premerlani. Object-Oriented Modeling and Design for Database Applications. Prentice Hall, 1998.

[BLAH01] M. Blaha. A Retrospective on Industrial Database Reverse Engineering Projects - Part 1 and Part 2, Working Conf. on Reverse Engineering, WRCE'01.

[BOJA99] C.E. Bojarczuk, H.S. Lopes et A.A. Freitas. Discovering comprehensible classification rules using genetic programming: a case study in a medical domain. Genetic and Evolutionary Computation Conf. (GECCO-99), p. 953-958, Orlando, USA, 1999.

[BOJA00] C.C. Bojarczuk, H.S. Lopes et A.A. Freitas. Genetic programming for knowledge discovery in chest pain diagnosis. IEEE Engineering in Medicine and Biology magazine - special issue on Data Mining and Knowledge Discovery, 19(4), p. 38-44, 2000.

[BOJA01] C.E. Bojarczuk, H.S. Lopes et A.A. Freitas. Data Miningwith constrained-syntax genetic programming: applications in medical data sets. Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-2001), a Workshop at Medinfo-2001, Londres, UK, 2001.

[BOSC97] A. Van den Bosch, T. Weijters, H. J. Van den Herik et W. Daelemans. When Small Disjuncts Abound, Try Lazy Learning : A Case Study. In Proc. of the Seventh Belgian-Dutch Conf. on Machine Learning, p. 109-118, 1997.

[BREI84] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Monterey, CA,Wadsworth Int. Group, 1984.

[BRIN97a] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In Proc.of the 1997 ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'97), p. 255-264, ACM Press, 1997.

[BRIN97b] S. Brin, R. Motwani et C. Silverstein. Beyond market baskets : generalizing association rules to correlations. ACM SIGMOD Int. Conf. on Management of Data, p. 265-276, Tuscon, Arizona, USA, 1997.

[BROW01] P. Brown. Object-Relational Database Development, A plumber's Guide. Prentice Hall, 2001.

[BUJA01] A. Buja et Y.-S. Lee. Data Miningcriteria for tree-based regression and classification. Proc. of the 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'2001), p. 27-36, 2001

[CARP93] C. Carpineto and G. Romano. GALOIS : An order-theoretic approach to conceptual clustering. Proc. of the 10th Int. Conf. on Machine Learning (ICML'90), p. 33-40, 1993.

[CARV99] D.R. Carvalho, B.C. Avila et A.A. Freitas. A hybrid genetic algorithm / decision tree approach for coping with unbalanced classes. 3rd Int. Conf. on the Practical Applications of Knowledge Discovery & Data Mining (PADD-99), p. 61-70, Londres. 1999.

[CARV00a] D.R. Carvalho et A.A. Freitas. A genetic algorithm-based solution for the problem of small disjuncts. Principles of Data Mining and Knowledge Discovery (PKDD'00), p. 345-352, Springer-Verlag, 2000.

[CARV00b] D.R. Carvalho et A.A. Freitas. A hybrid decision tree / genetic algorithm for coping with the problem of small disjuncts in fouille de données. Genetic and Evolutionary Computation Conf. (GECCO-2000), p. 1061-1068, Las Vegas, NV, USA, 2000.

[CASA83] M.A. Casanova and J.E.A. de Sa. Designing Entity-Relationship Schemes for Conventional Information Systems. Proc. of the 3td Int. Conf. on the ER Approach to Software Engineering, p. 265-277, Anaheim, California , 1983.

[CATT99] R. G. G. Cattel, D. Barry, M. Berler, J. Eastman, D. Jordan. C. Russel, O. Schadow, T. Stanienda and F. Velez. The Object Data Standard : ODMG 3.0. Morgan Kaufman Publishers, 1999.

[CHEE96] P. Cheeseman and J. Stutz. Bayesian classification (AutoClass) : Theory and results. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthutusamy Ed., Advances in knowledge Discovery and Data Mining, p. 153 – 180, AAAI Press, 1996.

[CHEN76] P. P.-S. Chen. The Entity-Relationship Model -- Toward a Unified View of Data. ACM Trans. Database Syst. 1(1), p. 9-36, 1976.

[CHIA94] R. H. L. Chiang, T-M Barron and V.C Storey. Reverse Engineering of Relational Databases : Extraction of an EER Model from a Relational Database. Data and Knowledge Engineering, 12, p. 107-142, 1994.

[CHIA95] R. H. L. Chiang. A knowledge-based system for performing reverse engineering of relational databases. Decision Support Systems 13, p. 295-312, 1995.

[CHIK90] E. J. Chikofsky and J. H. Cross. Reverse Engineering and Design Recovery : A Taxonomy. IEEE Software, p. 13-17, 1990.

[CHOE98] S.Choenni. On the suitability of genetic-based algorithms for data mining, Proc. of the Int. Workshop on Advances in Databases Technologies, Springer Verlag, 1998.

[CODD70] E. F. Codd. A Relational Model of Data for Large Shared Data Banks. Communications ACM, V13, 6, p. 377-387, 1970.

[COLL93] M. Collard. Un langage de requêtes déductif pour objets persistants, Rapport de thèse, Université de Nice-Sophia Antipolis, 1993.

[COLL94] M. Collard, N. Le Thanh. A Deductive Query Language for the Integration of different Programming Styles, 2nd European Joint Conf. on Engineering Systems Design and Analysis, p. 34-46, Londres, 1994.

[COLL97] M. Collard. Une méthode d'extraction de connaissances pour l'aide à la conception orientée objet, INFORSID, p. 88-97, Toulouse, 1997.

[COLL98] M.Collard, Mining Dependency Relationships for Object Oriented Modelling. Int. ICSC Symposium on Engineering of Intelligent Systems, p. 228-235, Tenerife, Espagne, 1998.

[COLL99] M. Collard, A. Barbar. Discovery of Synonymy Rules by Mining Queries. Int. Conf. on Computational Intelligence for Modelling Control and Automation, p. 122-130, Vienne, Autriche, 1999.

[COLL01a] M. Collard, A. Barbar. Semantic Extraction : a User-Driven Method. ISM'01, 4th Int. Conf. on Information Systems Modelling, p. 77-84, Czech Republic, 2001.

[COLL01b] M. Collard, D. Francisci, Evolutionary Data Mining: an overview of Genetic-based Algorithms. 8th IEEE Int. Conf. on Emerging Technologies and Factory Automation, ETFA'2001, p. 4-10, Antibes, France, 2001.

[COLL02a] Y. Collette et P. Siarry. Optimisation multiobjectif, Editions Eyrolles. 2002.

[COLL02b] M. Collard, A. Barbar, Mining Legacy Databases. ISE'02, Int. Symposium on Information Systems and Engineering, p. 120-125, San Diego, USA, 2002.

[COLL02c] M. Collard, A. Barbar. MoRE : An Object Model for Eliciting Relational Data Semantics. 2nd IEEE Int. Symposium on Signal Processing and Information Technology (ISPITT'02), Marrakech, 2002.

[COLL02d] M. Collard. EMA : An evolutionary method for Modelling by Mining Examples. ISE'02, Int. Symposium on Information Systems and Engineering, p. 110-114, San Diego, USA, 2002.

[COLL02e] M. Collard. Tutorial : An Overview of Database Reverse Engineering Methods. 2002 Summer Computer Simulation Conf., San Diego, USA, 2002

[COLL03] M. Collard, A. Cavarero. Une approche basée sur le comportement utilisateur pour la re-ingénierie de processus, Poster, IC'03, Ingénierie des Connaissances, Plate-forme de l'AFIA, Laval, 2003.

[CONG00] C. B. Congdon et E. F. Greenfest. Gaphyl : a genetic algorithm approach to cladistics. GECCO-2000 Workshop Programs. 2000.

[CRAV94] M. Craven and J. W. Shavlik. Using sampling and queries to extract rules from trained neural networks. Proc. of the 11th Int. Conf. on Machine Learning, p. 37-45, New Brunswick, NJ, Morgan Kaufmann, 1994.

[DAGA99] Dagnan I., Lee L. and Pereira F. Similarity-based models of word co-occurence probabilities. Machine Learning, 34 : 43-69, 1999.

[DANY93] A. P. Danyluk et F. J. Provost. Small disjuncts in action : learning to diagnose errors in the local loop of the telephone network. Proc.of the 10th Int. Conf. on Machine Learning (ICML'93), p. 81-88. 1993.

[DAS98] G. Das, H. Mannila and P. Ronkainen. Similarity of Attributes by External Probed. In R. Agrawal, P. Storloz and G. Piatetsky-Shapiro, Ed.. Proc. of the 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD'98), p. 23-29, New York, NY, USA, 1998. AAAI Press.

[DAS00] G. Das and H. Mannila. Context-based similarity methods for categorical attributes. Principles of Data Miningand Knowledge Discovery, 4th European Conf. (PKDD 2000) D.A. Zighed et al. Ed., p. 201-211, 2000.

[DAVI87] K. H. Davis, A. Arora. Converting a Relational database Model into an Entity-Relationship Model. Proc. of the 6th Int. Conf. on ER Approach, p. 243-256, New-York, 1987.

[DAVI00] K. H. Davis and P. H. Aiken. Data Reverse Engineering : A Historical Survey. Proc. of the 7th working Conf. on Reverse Engineering WCRE'00, 2000.

[DBMA02] DBMAIN 6.5 Reference manual. Université de Namur, 2002.

[DEB01] K. Deb. Multi-objective optimization using evolutionary algorithms. Wiley. 2001.

[DHAR00] V. Dhar, D. Chou et F. Provost. Discovering interesting patterns for investment decision making with GLOWER – a genetic learner overlaid with entropy reduction. Data Mining and Knowledge Discovery Journal 4(4), p. 251-280, 2000.

[DITT97] K. R. Dittrich and A. Geppert. Object-Oriented DBMS and Beyond. Proc. of SOFSEM'97 : Theory and Practice of Informatics, Milovy, Tchechoslovaquie, 1997.

[EMMA00] C. Emmanouilidis, A. Hunter et J. MacIntyre. A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. Proc. of the 2000 Congress on Evolutionary Computation (CEC'2000), p. 309-306, IEEE 2000.

[FABR99] C.C. Fabris et A.A. Freitas. Discovering surprising patterns by detecting occurrences of Simpson's paradox. Research and Development in Intelligent Systems XVI, p. 148-160. Springer-Verlag. 1999.

[FAYY96] U.M. Fayyad, G. Piatetsky-Shapiro and P. Smyth. Knowledge Discovery and Data Mining : Towards a Unifying Framework. In proceedings of the 2nd international conference on Knowledge Discovery and Data Mining (KDD'96), p. 82-88. AAAI Press, 1996.

[FAYY96] U. Fayyad, G. Piatetsky-Shapiro et P. Smyth. The KDD process for extracting useful knowledge from volumes of data. 1996.

[FIDE00] M.V. Fidelis, H.S. Lopes et A.A. Freitas. Discovering comprehensible classification rules with a genetic algorithm. Proc.Congress on Evolutionary Computation - 2000 (CEC-2000), p. 805-810. La Jolla, CA, USA. 2000.

[FLEU95] L. Fleury et Y. Masson. The intensity of implication, a measurement learning machine. IEA/AIE 95, Melbourne, 1995.

[FLOC95] I. W. Flockart, N. J. Radcliffe. GA-Miner : Parallel Data Miningwith Hierarchical genetic Algorithms, Final Report. EPCC-AIKMS-GA-Miner-Report 1.0, University of Edimburgh, 1995.

[FLOR01] A. Flory et F. Laforest. Les bases de données relationnelles. Editions Economica – 2001.

[FOGE00] D.B. Fogel. Introdution to Evolutionary Computation. Evolutionary Computation1- Basic Algorithms and Operators. T.Back, D.B.Fogel and T.MIchalewicz Ed. Institute of Physics Pub, Bristol and Philadelphia, 2000.

[FONK92] M. M. Fonkam and W. A. Gray. An Approach to Eliciting the Semantics of Relational Databases. Proc. of the 4th Int. Conf. on Advanced Information Systems Engineering, CAiSE'92, p. 463-480, Springer-Verlag, 1992.

[FRAN01] D. Francisci, M. Collard, Fouille de données et algorithmes évolutionnaires, Rapport interne, I3S/RR-2001-08-FR, 2001.

[FRAN03a] D. Francisci, M. Collard, Multi-Criteria Evaluation of Interesting Dependencies according to a Data Mining Approach, CEC'03, 2003 Congress on Evolutionary Computation, Canberra, Australie, 2003, à paraître.

[FRAN03b] D. Francisci, M. Collard, Optimizing rule quality in a fouille de données context, CIRAS'03, 2nd Int. Conf. on Computational Intelligence, Robotics and Autonomous Systems, Singapour, 2003, à paraître.

[FRAN03c] D. Francisci, M. Collard, Towards a multi-objective rule selection in data mining : an experimental approach to compare measures. Applied Mathematics, Operational Research and Optimization, Session ""Metaheuristics and multi-criteria optimisation"", CESA'03, Lille, France, 2003, accepté.

[FRAN04] D. Francisci. Techniques d'optimisation pour l'extraction automatique de connaissances. Thèse de l'université de Nic-Sophia Antipolis, 2004, à paraître.

[FREI96] A.A. Freitas et S.H. Lavington. Speeding up knowledge discovery in large relational databases by means of a new discretization algorithm. R. Morrison & J. Kennedy Ed. LNCS 1094, Advances in Databases, BNCOD, p. 124-133. Springer-Verlag. 1996.

[FREI98] A.A. Freitas. On objective measures of rule surprisingness. Principles of Data Mining and Knowledge Discovery (PKDD'98), Nantes, France, 1998.

[FREI99] A.A. Freitas. A genetic algorithm for generalized rule induction. Advances in Soft Computing - Engineering Design and Manufacturing, p. 340-353, Springer-Verlag. 1999.

[FREI01] A. A. Freitas. Understanding the crucial differences between classification and discovery of association rules - a position paper. ACM SIGKDD Explorations (ACM 2000), 2(1), p. 65-69, 2000.

[FREI01] A. A. Freitas. Understanding the Crucial Role of Attribute Interaction in Data Mining, Artificial Intelligence Review, 16(3), p. 177-199, 2001.

[FREI02] A. A. Freitas. Data Mining and knowledge discovery with evolutionary algorithms. Natural computing series. Springer 2002.

[GARC93] M. Garcia-Solaco, M.Castellanos, and F. Saltor. Discovering interdatabase resemblance of classes for interoperable databases. Proc. of the 3rd Int. Workshop on Research Interests in Data Engineering, Interoperability in Multidatabase Systems, RIDE-IMS-93, Vienna, Austria, 1993.

[GELF91] S. B. Gelfand, C. S. Ravishankar, and E. J. Delp. An iterative growing and pruning algorithm for classification tree design. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991.

[GEUD97] G. Guedj. AMC* Designor - Mise en oeuvre de Merise. conception d'application client-serveur, Editions Eyrolles, 1997.

[GION99] A. Gionis, P. Indyk and R. Motwani. Similarity Search in High Dimensions via Hashing. Proc.of the 25th Int. Conf. on Very Large Data Bases (VLDB'99), p. 518-529, Edinburgh, Scotland, UK, 1999. Morgan Kaufmann.

[GOLD89] D. E. Goldberg. *Genetic* Algorithms in Serach, Optimization, and Machine Learning. Addison-Wesley, Reading, MA, 1989..

[GOOD91] R. M. Goodman et P. Smyth. Rule induction using information theory. G. Piatetsky – Shapiro et W. J. Frawley Ed., Knowledge Discovery in Databases. MIT Press. 1991.

[GRAS01] R. Gras. L'implication statistique – Nouvelle méthodé exploratoires de données. Editions " La pensée sauvage ", 2001

[GREF95] J.J.Grefenstette. Predictive models using fitness distributions of genetic operators, Foundations of Genetic Algorithms, San Mateo, CA, 1995.

[GUIL98] S. Guillaume, F. Guillet et J. Philippé. Improving the Discovery of Association Rules with Intensity of Implication. PKDD 1998, p. 318-327, 1998.

[GUIL01] S. Guillaume. L'intensité d'inclination : une généralisation de l'intensité d'implication ordinale. Actes des 8ièmes journées de la société francophone de classification (SFC'01). Université de Antilles-Guyane, 2001.

[HAIN91] J. L. Hainaut. Database Reverse Engineering : Models, techniques and strategies. Proc. of the 10th Int. Conf. on Entity-Relationship Approach, p. 729-741, San Mateo, 1991.

[HAIN93] J. L. Hainaut, C. Tonneau, M. Joris and M. Chandelon. Schema Transformation Techniques for Database Reverse Engineering. Proc. of the 12th Int. Conf. on ER Approach, 1993.

[HAIN94] J. L. Hainaut, C. Tonneau, M. Joris and M. Chandelon. Transformation-based Database Reverse Engineering. Proc. Of the 13th Int. Conf. On ER Approach, p. 364-375, Manchester, UK, 1994.

[HAIN95] J. L. Hainaut, V. Engelbert, J. Henrard, J. M. Hick and D. Roland. Requirements for Information System Reverse Engineering Support. Proc. of the IEEE Working Conf. on Reverse Engineering, p. 136-145, Toronto, Canada, IEEE Computer Society Press, 1995.

[HAMI97] H.J. Hamilton, N. Shan, and W. Ziarko. Machine learning of credible classifications. Advanced Topics in Artificial Intelligence, Tenth Australian Joint Conference on Artificial Intelligence (AI'97), p.330-339, Perth, Australia, 1997.

[HAND01] D. Hand, H. Mannila, and P. Smyth. Principles of Data Mining. MIT Press, 2001.

[HART75] J. A. Hartigan. Clustering Algorithms. John Wiley & Sons, 1975.

[HERI01] D. HÉRIN, D.A. ZIGHED. Actes de la conférence EGC 2002: Extraction et gestion des connaissances, Hermes, 2002

[HICK98] J. M. Hick et J. L. Hainaut. Maintenance et évolution d'applications de bases de données. INFORSID, Villeurbanne, 1998.

[HILD99a] R. J. Hilderman et H. J. Hamilton. Heuristics for Ranking the Interestingness of Discovered Knowledge. Proc. of the 3rd Pacific-Asia Conf. on Methodologies for Knowledge Discovery and Data Mining (PAKDD'99), p. 204-209, Beijing, China, 1999.

[HILD99b] R. J. Hilderman et H. J. Hamilton et B. Barber. Ranking the Interestingness of Summaries from Data Mining Systems. Proc. of the 12th Int. Florida Artificial Intelligence Research Symposium (FLAIRS'99) , p. 100-106, Orlando, USA, 1999.

[HILD01] R. J. Hilderman et H. J. Hamilton, H.J. Evaluation of Interestingness Measures for Ranking Discovered Knowledge. Proc.of the 5th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'01) p. 247-259., Hong Kong, 2001.

[HOLL75] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.

[HOLT89] R. C. Holte, L. E. Acker et B. W. Porter. Concept Learning and the Problem of Small Disjuncts. Proc.of the Eleventh Int. Joint Conf. on Artificial Intelligence, p. 813-818, 1989.

[HUHT99] Y. Huhtala, J. Kärkkänen, P. Porkka and H. Toivonen. Efficient Discovery of Functional and Approximate Dependencies Using Partitions. Technical Report C-1997-79.

[HUSS00] F.Hussain, H.Liu and H.Lu. Relative Measure for Mining Interesting Rules. Proc. of the 4th European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'00), p. 432-439, Lyon, France, 2000.

[IBM96] Int. Business Machines, IBM intelligent Miner, User's guide, version 1, release 1, 1996.

[JAIN99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering : A survey. ACM Comput. Surv., 31 : 264-323, 1999.

[JOHA90] P. Johannesson and K. Kalman. A Method for Translating Relational schemas into Conceptual Schemas. Proc. of the 8th Int. Conf. on ER Approach, p. 279-283, 1990.

[JOHA94] P. Johannesson. A method for transforming Relational Schemas into Conceptual Schemas. IEEE Int. Cong. On Data Engineering (ICDE), Los Alamitos, p. 190-201, 1994.

[JONE95] T.Jones , S.Forrest. Genetic Algorithms and Heuristic Search, Santa Fe Institute Technical Report 95-02-021. Santa Fe Institute, 1995.

[JORI92] M. Joris, R. V. Hoe, J. L. Hainaut, E. Cardon, M. Chandelon, C. Tonneau, P. Verscheure, F. Bodart, F. Vandamme and M. Vanwormhoudt. Phenix : Methods and Tools For Database Reverse Engineering. Proc. of the 5th international conference of Software engineering and Applications, p. 541-551, Toulouse, 1992.

[JOSH02] M. Joshi. On Evaluating Performance of Classifiers for Rare Classes. ICDM 02, Maebashi City, Japan, 2002.

[JOUR02] L. Jourdan, C. Dhaenens, E.G. Talbi. A genetic algorithm to exploit genetic data. Volume Evolutionary Computation and Bioinformatics, D. Corne and G. Fogel Ed., Evolutionary Computation and Bioinformatics Morgan Kaufmann, p. 297--316, 2002.

[KALM91] K. Kalman. Implementation and Critic of an algorithm which maps a Relational Database to a Conceptual Model. Proc. of the 3th Int. Conf. on Advanced Information Systems Engineering – CaiSE'91, p. 393-415, 1991.

[KANT92] M. Kantola, H. Mannila, K. J. Raïhä, and H. Siirtola. Discovering functional and inclusion dependencies in relational databases. Int. Journal of Intelligent Systems, 7 : 591-607, 1992.

[KARY99] G. Karypis, E. H. Han, and V. Kumar. CHAMELEON : A hierarchical clustering algorithm using dynamic modeling. Computer, 32 : 68-75, 1999.

[KASH96] V. Kashyap, A.Sheth. Semantic and schematic similarities between database objects : a context-based approach. VLDB Journal, Vol. 5, p 276-304, 1996

[KAUF90] L. Kaufman and P. J. Rousseuv. Finding Groups in data : An Introduction to Cluster Analysis. John Wiley & Sons, 1990.

[KIM00] Y. Kim, W. N. Street et F. Menczer. Feature selection in unsupervised learning via evolutionary search. Proc. of the 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'2000), p. 365-369, ACM 2000.

[KLEM94] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen et I. Verkamo. Finding interesting rules from large sets of discovered association rules. Proc. of the Third Int. Conf. on Information and Knowledge Management (CIKM'94, p. 401-407, 1994.

[KONO94] I. Kononenko. Estimating attributes : Analysis and extensions of RELIEF. Proc.ECML-94, (F.Bergadano, L.de Readt Ed.), Springer Verlag, p. 171-182, Catania, Sicile, 1994.

[KOSA92] J.R. Kosa. Genetic Programming: on the Programming of Computers by Means of Natural Selection. Cambridge, MA: MIT Press, 1992.

[KULK95] K. Kulkarni, M. Carey, L. DeMichiel, N. Mattos, W. Hong, M. Ubell, A. Nori, V. Krishnamurthy, and D. Beech. Introducing Reference Types and Cleaning Up SQL3's Object Model. Int. Organization for Standardization, 1995.

[KUNT02] P. Gras, P. Kuntz, R. Coutuier et F. Guillet. Une version entropique de l'intensité d'implication pour les corpus volumineux. Extraction des connaissances et apprentissage, 1(4), p. 69-80. 2002.

[KWED 98] W. Kwedlo et M. Kretowski. Discovery of decision rules from databases : an evolutionary approach. Principles of Data Mining and Knowledge Discovery, Nantes, France, 1998.

[LALL02] S. Lallich et O. Teytaud. Evaluation et validation de l'intérêt des règles d'association. Rapport de recherche pour le groupe de travail GafoQualitée de l'action spécifique STIC fouille de bases de données, ERIC, Université Lyon 2, 2002.

[LEE97] L. J. Lee. Similarity-Based Approaches for Natural Language Processing. PH.D. Thesis, Harvard University, Cambridge, 1997.

[LENC02] P. Lenca, P. Meyer, B. Vaillant et P. Picouet. Aide multicritère à la décision pour évaluer les indices de qualité des connaissances – Modélisation des préférences de l'utilisateur. Rapport de recherche pour le groupe de travail GafoQualitée de l'action spécifique STIC fouille de bases de données, département IASC, ENST Bretagne, 2002.

[LIM00] T.-S. Lim, W.-Y. Loh et Y.-S. Shih. A comparison of prediction accuracy, complexity and training time of thirty-three old and new classification algorithms. Machine Learning 40, p. 203-228,. 2000.

[LIU96] B. Liu et W. Hsu. Post-analysis of learned rule. Proc. of the 1996 National Conf. of the American Association for Artificial Intelligence (AAAI'96), AAAI Press, 1996.

Nagoya, Japan, 1997.

[LIU98] B. Liu, W. Hsu et Y. Ma. Integrating Classification and Association Rule Mining. Proc. of the Fourth International Conference on Knowledge Discovery and Data Minig (KDD98), New York, USA, 1998.

[LIU99] B. Liu, W. Hsu et Y. Ma,. Pruning and Summarizing the Discovered Associations. Proc.of the ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining, San Diego, CA, USA, 1999.

[LIU01] B. Liu, Y. Ma et C.-K. Wong. Classification Using Association Rules : Weaknesses and Enhancements. Data Mining for scientific applications. 2001.

[LOEV47] J. Loevinger. A systemic approach to the construction and evaluation of tests of ability. Psychological monographs, 61(4), 1947.

[LOP97] H. S. Lopes, M. S. Coutinho et W. C. Lima. An evolutionnary approach to simulate cognitive feedback learning in medical domain. E. Sanchez, T. Shibata et L. A. Zadeh Ed. Genetic Algorithms and Fuzzy Logic Systems, p. 193-207, Singapore: World Scientific, 1997.

[MACQ67] J. MacQueen. Some methods for classification and analysis of multivariate observations. Proc. 5th Berkeley Symp. Math. Statist, Prob., 1 : 281-297, 1967.

[MAND91] B. Manderick, M. de Weger, and P. Spiessens. The genetic algorithm and the structure of the fitness landscape. Proc. of the Fourth Int. Conf. on Genetic Algorithms, 1991.

[MANN87] H. Mannila and K. J. Räihä. Dependency Inference. Proc. of the 13th Int. Conf. on Very Large Databases (VLDB), Brighton, England, p. 155-158, Morgan Kaufmann, 1987.

[MANN92] H. Mannila and K. J. Räihä. The Design of Relational Databases, Wokingham, England : Addison-Wesley, 1992.

[MARK90] V. M. Markowitz and J. A. Makowsky. Identifying Extended Entity-Relationship Object Structures in Relational Schemas. IEEE Transactions on Software Engineering, 16(8): 777-790, 1990.

[MEND01] R.R.F. Mendes, F.B. Voznika, A.A. Freitas et J.C. Nievola. Discovering fuzzy classification rules with genetic programming and co-evolution. Principles of Data Miningand Knowledge Discovery (PKDD 2001), p. 314-325. Springer-Verlag. 2001.

[MOEN00] P.Moen. Attribute, Event Sequences and Event Type Similarity Notions for Data Mining, PhD Thesis. Report A-2000-1, Université d'Helsinki, Finland.

[NAKH97] G. Nakhaeizadeh, A. Schnabl. Development of Multi-Criteria Metrics for Evaluation of Data Mining Algorithms. KDD 1997.

[NAVA87] S. Navathe and A. Awong. Abstracting Relational and Hierarchical Data with a Semantic Data Model. Proc. of the 6th Int. Conf. on the ER Approach, p. 277-305, 1987.

[NG98] R.T. Ng, Laks V. S. Lakshmanan, J.Han, and A.Pang. Exploratory mining and pruning optimizations of constrained association rules. Proc. ACM SIGMOD, p. 13-24, 1998.

[NIER89] O. Nierstraz. A Survey of Object-Oriented Concepts. In W. Kim and F. H. Lochovsky Ed., Object-Oriented Concepts, Databases, and Applications. ACM Press, New York, 1989.

[NODA99] E. Noda, A.A. Freitas et H.S. Lopes. Discovering interesting prediction rules with a genetic algorithm. Congress on Evolutionary Computation (CEC-99), p. 1322-1329. Washington D.C., USA. Juillet 1999.

[ORAC00] Oracle Designer 6i New Features. An Oracle Technical White Paper. http://www.Oracle.com, 2000.

[PARS89] K. Parsaye.et al. Intelligent databases. Toronto, John Wiley & Sons , 1989.

[PATH00] P. Pathak, M. Gordon et W. Fan. Effective information retrieval using genetic algorithms based matching functions adaptation, HICSS 2000.

[PETI94] J. M. Petit, J. Kouloumdjian, J. F. Boulicaut and F. Toumani. Using Queries to Improve Database Reverse Engineering. Int. Conf. on the Entity-Relationship Approach (ERA), Manchester, p. 369-386, 1994.

[PETI95] J. M. Petit, F. Toumani and J. Kouloumdjian. Relational Database Reverse Engineering : a Method Based on Query Analysis. Int. Journal of Cooperative Information Systems, vol. 4, no 2, 3, p. 287-316, 1995.

[PETI96] J. M. Petit, F. Toumani, F. F. Boulicaut and J. Kouloumdjian. Towards the Reverse Engineering of Denormalized Relational Databases. Proc. of the 12th Int. Conf. on Data Engineering, IEEE, New Orleans, USA, 1996.

[PIAT91] G. Piatetsky-Shapiro. Discovery, analysis and presentation of strong rules. G. Piatetsky – Shapiro and W. J. Frawley Ed., Knowledge Discovery in Databases, MIT Press. 1991.

[PIAT94] G. Piatetsky-Shapiro, C.J. Matheus. The interestingness of deviations. Knowledge Discovery in Database Workshop, 1994.

[PIAT00] Measuring Lift Quality in Database Marketing. G. Piatetsky-Shapiro and S. Steingold Ed., SIGKDD Explorations, 2000.

[PREM94] W. Premerlani and M. Blaha. An approach for Reverse Engineering of Relational Databases. Communications of the ACM, 37(5), p. 42-49, 1994.

[QUIN93] J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.

[QUIN96] J. R. Quinlan. Bagging, boosting, and C4.5. Proc. 13th Nat. Conf. Artificial Intelligence (AAAI'96), p. 725-730, Portland, OR, 1996.

[RADC94] N. J.Radcliffe, P.D. Surry. Co-operation through Hierarchical Competition in Genetic Data Mining, EPCC-TR94-09, 1994.

[RAMA97] C. Ramanathan. Providing Object-Oriented Access to Existing Relational Databases. Ph.D. thesis, Mississipi State University, 1997.

[ROBN99] M. Robnik-Sikonja et I. Kononenko. Attribute dependencies, understandability and split selection in tree based models. Int. Conf. on Machine Learning ICML-99, p. 27-29, Bled, 1999.

[RODD01] J. F. Roddick et S. Rice. What's interesting about cricket ? On thresholds and anticipation in discovered rules. SIGKDD Explorations, 2001.

[RODG89] U. Rodgers. Denormalization : Why, What, and How ?. Database Programming and Design. p. 46-53, 1989.

[ROMA02] W. Romao, A.A. Freitas et R.C.S. Pacheco. A Genetic Algorithm for Discovering Interesting Fuzzy Prediction Rules: applications to science and technology data. Genetic and Evolutionary Computation Conf. (GECCO-2002), New York, 2002.

[ROSE96] Helge Rosé,Werner Ebeling, and Torsten Asselmeyer. The density of states - a measure of the difficulty of optimisation problems. Parallel Problem Solving from Nature, p. 208-217, 1996.

[SAHA99] S. Sahar. Interestingness via what is not interesting. SIGKDD 99..

[SANT00] R. Santos, J.C. Nievola et A.A. Freitas. Extracting comprehensible rules from neural networks via genetic algorithms. IEEE Symp. on Combinations of Evolutionary Computation and Neural Networks (ECNN-2000), p. 130-139. San Antonio, TX, USA, 2000.

[SAVN93] I. Savnik and P. A. Flach. Bottom-up induction of functional dependencies from relations. In G. Piatetsky-Shapiro Ed., Proc. of the Third Workshop on Knowledge Discovery in Databases, Washington, USA, p. 174-185, 1993.

[SCHW] HP Schwefel. Advantages (and disadvantages) of evolutionary eomputation over other approaches.. Evolutionary Computation, Thomas Bäck Ed., Institute of Physics Pub, 2000.

[SEBA88] M.Sebag, M.Schoenauer. Generation of rules with certainty and confidence factors from incomplete and incoherent learning bases. Proc. of the European Knowledge Acquisition Workshop, EKAW'88.

[SHET93] A. Sheth and V. Kashyap. So far (schematically) yet, so near (semantically). In D. K. Hsiao, E. J. Neuhold, and R. Sacks-Davis Ed., Interoperable database systems, p. 283-312, Elsevier Science Publishers B. V., 1993.

[SHOV93] P. Shoval and N. Shreiber. Database reverse engineering : From the Relational to the Binary Relationship Model. Data and Knowledge engineering, p. 293-315, 1993.

[SIGN94] O. Signore, M. Loffredo, M. Gregori and M. Cima. Reconstruction of ER Schema from Database Applications : a Cognitive Approach. Proc. of the 13th Int. Conf. Approach, p. 387-402, Manchester UK, 1994.

[SILB96] A. Silberschatz, A.Tuzhilin. What makes patterns interesting in knowledge discovery, A. Silberschatz and A. Tuzhilin. IEEE Transactions on Knowledge and Data Eng., 8(6):970--974, 1996.

[SMIT91] P.Smith, H.M. Goodman. Rule induction using information theory. Knowledge Discovery in Databases. G. Piatetsky – Shapiro et W. J. Frawley Ed., MIT Press. 1991.

[SOUT96] C. Soutou. Extracting N-ary Relationships Through Database Reverse Engineering. Proc. of the 15th Int. Conf. on Conceptual Modeling (ER'96), p. 392-405, 1996.

[SRIN95] N.Srinivas and K.Deb. Multiobjective optimization using non dominated sorting in genetic algorithms. EvolutionaryComputation, 2(3):221–248, 1995.

[STON96] M.Stonebraker, D.Moore. Object-Relational DBMSs. Morgan Kaufmann Pub., 1996.

[STOR97] V. C. Storey, R. H. L. Chiang, D. Dey, R. C. Goldstein and S. Sundaresan. Database Design With Common Sense Business Reasoning and Learning. ACM Transactions on Database Systems, p. 471-512, 1997.

[TARI96] Z. Tari, O. Bukhres, J. Stokes and S. Hammoudi. The Reengineering of Relational Databases based on Key and Data Correlations. IFIP 1996. p. 183-214.

[TEOR86] T. J. Teorey, Y. Dongqing and J. P. Fry. A logical design methodology for relational databases using the Extended Entity Relationship Model. ACM Comput. Surveys 18(2), p.197-222.

[TEUS00] T. Teusan, G. Nachouki, H. Briand et J. Philippe. Discovering Association Rules in Large, Dense Databases. Proc. 4th European Conf., p. 638-645, PKDD-2000. Lyon, France.

[TILL93] S. R. Tilley, H. A. Muller, M. J. Withney and K. Wong. Domain-Retargetable Reverse Engineering. Proc. of EEE Working Conf. on Software Maintenance, p. 142-151, 1993.

[TOMA02] M. Tomassini, L. Vanneschi, F.V. Fernández, G.G. Gil. Experimental Investigation of Three Distributed Genetic Programming Models. PPSN 2002, p.641-650, 2002.

[TURN01] P. Turney. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. Proc. of the Twelfth European Conf. on Machine Learning (ECML-2001).

[VENT93] G.Venturini. SIA : A supervised inductive algorithm with genetic search for learning attributes based concepts. Proc. of the European Conf. on Machine Learning, p. 280-296, 1993.

[WANG99] K. Wang, S. Zhou, and S. C. Liew. Building Hierarchical classifiers using class proximity. Int. Conf. Very Large Data Bases (VLDB'99), p. 363-374, Edinburgh, UK, 1999.

[WATT96] I. Wattiau and J. Akoka. Reverse Engineering of Relational Database Physical Schemas. Proc. of the 15th Int. Entity Relationship Conf., Cottbus, Allemagne, 1996.

[WEIS98] G. M. Weiss et H. Hirsh. The Problem with Noise and Small Disjuncts. In Proc.of the Fifteenth Int. Conf. on Machine Learning. Morgan Kaufmann Publishers. San Francisco, CA, 574-578.1998.

[WEIS00] G. M. Weiss et H. Hirsh. A Quantitative Study of Small Disjuncts. Proc. of the Seventeenth National Conf. on Artificial Intelligence, Austin, Texas, 2000.

[WEIS01] G. M. Weiss et F. Provost. The effect of class distribution on classifier learning. Technical report ML-TR-43, Department of Computer Sciences, Rutgers University, 2001.

[WEKA] Weka Software, Université de Waikato, Nouvelle Zélande, http://www.cs.waikato.ac.nz/ml/weka/

[WITT99] I. H. Witten et E. Frank. Data Mining– Practical machine learning tools and techniques with Java implementations. Morgan Kaufmann Publishers. 1999.

[WOLP96] D.H.Wolpert and William G. Macready. No Free Lunch Theorems for Search. Santa Fe Institute, Working Papers, 1996.

[ZAKI00] M.J. Zaki. Generating non-redundant association rules. Proc. of the 6th ACM SIGKDD Intl. Conf., Boston, MA, 2000.

[ZIGH98] D. A. Zighed, S. Rabaseda, and R. Rakotomalala. Fusinter : a method for discretization of continous attributes for supervised learning. Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(33) : 307-326, 1998.

[ZIGH00] D. A. Zighed and R. Rakotomalala. Graphes d'induction – Apprentissage et Fouille de données. Editions Hermès, Paris, 2000.