



N° d'ordre :

UNIVERSITE DE M'SILA
FACULTE DES MATHÉMATIQUES ET DE
L'INFORMATIQUE
Département d'Informatique

MEMOIRE

Présenté pour l'obtention du diplôme de d'ingénieur d'état

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Systèmes d'informations avancés

Par :

- BEN NEDJEMA BELKACEM

- BENLAITER AMEURE

SUJET

**EXTRACTION DES REGLES D'ASSOCIATION POUR
L'AIDE A LA DECISION
ETUDE DE CAS : PHARMACIE**

Soutenu publiquement le : devant le jury composé de :

Mr.

Mr. TAHAR MEHENNI

Mr.

Mr.

Mr.

Université de M'sila

Université de M'sila

Université de M'sila

Université de M'sila

Université de M'sila

Président

Rapporteur

Examineur

Examineur

Examineur

Promotion : 2011 /2012

Remerciement

*Tout d'abord, nous présentons nos remerciements les plus sincères à notre encadreur Mr **Tahar Mehenni**, pour sa disponibilité et son encadrement, ainsi que son soutien et ses conseils tout au long de l'année. C'est grâce à-lui que nous avons pu mener à bien ce travail. Nous tenons ensuite à remercier chacun des membres du jury pour nous avoir fait l'insigne honneur d'accepter d'examiner notre travail. Nos remerciements et notre et notre gratitude vont aux enseignants du département informatique. Nous remercions également toute personne ayant contribué à notre éducation et notre formation.*

Dédicaces

*Nous dédions ce modeste travail:
À nos chers parents, pour leurs soutiens, leurs patiences et
leurs amours
À toutes nos familles
À tous nos amis
Et à tous nos enseignants*

BELKACEM , AMEURE

ملخص

إن قواعد التجميع تستخدم بقوة في قواعد البيانات لاتخاذ القرارات. ومدى صلاحيتها يعتمد على مقاييس الجودة وأكثرها استعمالا معياري الدعم والثقة يطبقان على قاعدة بيانات المعاملات، التي هي من «90% من الزبائن الذين يشترون <<Beconase>> و <<Primalan>> يشترون أيضا <<Aspegic adultes>>»، علما أن 75% من الزبائن يشترون هذه الأدوية في عملنا هذا، اهتمامنا باستخراج المعلومات باستخدام تقنية البحث بقواعد التجميع في قاعدة بيانات خاصة بمعاملات صيدلية وذلك باعتماد على خوارزمية "Apriori" التي تسمح لنا بإيجاد قائمة الأدوية الأكثر تكرارا (مبيعا) وقواعد الارتباط فيما بينها وذلك بقياس الارتباط الايجابي و الاستلزام المنطقي بين هذه النتائج تستخدم في اتخاذ القرارات (التنظيمية، تسيير الطلبات، تسيير المخزون ... الخ).

الكلمات المفتاحية: التنقيب في المعطيات، قواعد التجميع، خوارزمية "Apriori"، مجموعة المواد الأكثر تكرارا

ABSTRACT

Association rules reveal non trivial patterns, potentially useful for decision making, from huge databases. Their interestingness is assessed by means of quality measures, the most used of which are being the support and the confidence. A standard example of an association rule for a transactional database is of the form: « 90% of consumers who bought <<Beconase>> and <<Primalan>> also bought <<Aspegic adultes>>, given 75% of consumers actually bought theses medicaments ».

In this work, is interested in the Knowledge Discovery in Databases (KDD) by using the research technique of association rules in a transaction database of a pharmacy with the help of Apriori algorithm that can find the lists of products frequent and the rules related to one another by measuring the positive dependence, and the logical implication between the premise and therefore these results are used to help decision and allow to make several interesting activities (organization, ordering, inventory management etc.).

Keywords: data mining, associations rule, frequents itemsets, a-priori algorithm.

RESUME

Les règles d'association révèlent des régularités non triviales et potentiellement utiles pour l'aide à la décision, dans des bases de données. Leur validité est évaluée par des mesures de qualités dont les plus utilisées sont le support et la confiance. Pour une base de données transactionnelle d'une pharmacie, elles sont du type « 90% des clients ayant acheté du <<Beconase>> et du <<Primalan>> ont également acheté du <<Aspegic adultes>>, sachant que 75% des clients ont achetés ces médicaments ».

Dans ce travail, s'intéresse par l'Extraction de Connaissances dans les Données (ECD) par l'utilisation de la technique de recherche de règles d'association dans une base de données transactionnelle d'une pharmacie à l'aide d'algorithme Apriori que permet de trouver les listes des produits fréquentes et les règles de corrélation entre eux par mesure de la dépendance positive, et l'implication logique entre la prémisse et le conséquent ces résultats sont utilisés pour l'aide à décision et permettre de faire plusieurs actions intéressants (organisation, les commandes, gestion de stocketc.).

Mots Clés: fouille de données, règles d'association, algorithme a-priori, ensemble des produits fréquentes.

Sommaire

INTRODUCTION GÉNÉRALE	IV
-----------------------------	----

CHAPITRE I. DATA MINING

I. Data mining.....	04
I. 1. Origine et émergence du concept de DataMining.....	04
I.2. Définition générale.....	04
I.3. Principe et spécificité.....	06
I.3.1. Principe.....	06
I.3.2. Le data mining est un processus	06
I.3.3. Méthode.....	07
I.3.4. Anecdotes.....	07
I.4. Algorithmes.....	08
I.4.1. Les méthodes non- supervisées.....	08
I.4.2. Les méthodes supervisées.....	09
I.4.3. Les méthodes de réduction de données.....	09
I.4.4. Pourquoi tant d'algorithmes ?.....	10
I.5.le Data mining et statistique.....	10
I.5.1.Quelques repères	10
I.5.2. Analyse des données	11
I.5.3.data mining	11
I.6.Etapes de création d'un projet data mining	11
I.7. Processus Data Mining.....	12
I.8.Synthèse des besoins.....	13
I.8.1.Synthèse des problématiques associées au DataMining.....	15
I.9.Quels logiciels utilise t'on en data mining?.....	16
I.10. Les besoins et les attentes des utilisateurs	16
I.10.1. Un exemple paradigmatique	16
I.10.2. La prédiction, premier axe de recherche	18
I.10.3. Second axe, la création de typologies.....	19

I.11.Conclusion	21
-----------------------	----

CHAPITRE II. REGLES D'ASSOCIATION

II. Règles D'association :.....	23
II. 1. Définition d'une règle d'association :.....	23
II. 2 Concepts et définition :.....	23
II.3 Utilité des règles d'associations:.....	24
II.4 Signification d'une règle d'association :.....	25
II.5 Actionnabilité des règles d'association :.....	25
II.5.1 Terminologie et notations :.....	26
II.5.2 Règles d'association :.....	26
II.5.3Algorithmes:.....	26
II.5.3.1 La Recherche des Itemsets Fréquents :.....	28
II.5.3.2 La Génération des Règles d'Association :.....	31
II.6 Mesures de qualité :.....	33
II.6. 1 Mesures objectives :.....	34
II.6. 1.1 Support et confiance (les limites) :.....	36
II.6. 1.2Le lift (ou l'intérêt) :.....	37
II.6.2 Mesures subjectives :.....	38
II.6.3 Mesures sémantiques :.....	38
II.7. Conclusion:.....	41

CHAPITRE III .L'ALGORITHME A-PRIORI

III.1. introduction:.....	43
III.2. Principe des algorithmes existants pour la recherche de règles:.....	44
III.3. La notation:.....	44
III.4. L'algorithme Apriori:.....	46
III.4. 1.Fonctionnement :.....	47
III.4. 2.Les procédures d'algorithme A-Priori:.....	48
III.4. 2.1.La procédure APrioriGen:.....	48
III.4. 2.2.Procedure Close, notion d'itemset fermé:.....	49
III.4. 2.2.1.Treillis des ensembles fermés:.....	50
III.4. 2.2.2.L'Algorithme Close:.....	50
III.4. 2.3.Procédure de génération des règles d'association à partir des EIF:.....	52
III.4. 2.3.1Réduction du nombre de règles :.....	52
III.4. 2.3.2.Ensemble de items fréquents:.....	53
III.5.Exemple détaillé de l'algorithme Apriori:.....	55
III.5.Complexité de base de la génération d'itemsets fréquents:.....	58
III.6.Améliorations proposées à Apriori :.....	59
III.7.conclusion :.....	62

CHAPITRE IV.CONCEPTION ET MISE EN ŒUVRE

IV.1. Introduction.....	64
IV.2. conception :.....	65
IV.2.1. Analyse des besoins:.....	65

IV.2.2. Spécification fonctionnelle :	65
IV.2.2.1. Les acteurs du système :	65
IV.2.2.2. Détermination des cas d'utilisation :	66
IV.2.2.3. Diagramme de séquence de cas d'utilisation « Extraction » :	67
IV.2.2.4. Diagramme de séquence de cas d'utilisation « Résultats » :	68
IV.3. Mise en oeuvre :	69
IV.3.1. Outils de développement :	69
IV.3.2. Description des différentes classes :	70
IV.3.3. Présentation de l'application :	72
IV.4. Conclusion :	77
Conclusion générale :	78
Bibliographie :	80

Liste des Tableaux

TAB.II.1 : Règles d'association générées à partir de deux variables X et Y	27
TAB.II.3 : Exemple : les confiances des différentes règles.....	34
TAB.II.4 : Exemples de mesures d'utilité.....	39
TAB.III.1 : Principe de l'algorithme Apriori.....	48
TAB.III.2 : La procédure AprioriGen.....	49
TAB.III.3 : ensemble mots fréquents $e = \{ABC\}$	52
TAB.III.4 : Procédure Gen- règles.....	54
TAB.IV.1 : Les acteurs du système.....	65
TAB.IV.2 : Les principales classes de notre système.....	70

Liste des figures

Figure I. 1 : Processus Data Mining CRISP-DM.....	12
Figure I. 2 : le schéma du grandes fonctionnalités de « miner » les données.....	14
Figure II.1 : Diagramme de Venn de l'ensemble X dans celui des transactions T	27
Figure II.2 : Historique des algorithmes d'extraction d'itemsets fréquents.....	28
Figure II.3 : Itemsets fréquents, fermés et maximaux.....	30
Figure II.4 : Treillis d'itemsets.....	31
Figure II.6 : Treillis partiel composé des itemsets fréquents.....	32
Figure .II.5 : Exemple : les supports des itemsets.....	33
Figure II.7 : Catégorisation des mesures d'intérêts des règles d'association.....	35
Figure II.8 : Rôles des mesures d'intérêts dans le processus d'ECD.....	36
Figure III.1 : Exemple d'une base de données transactionnelle contenant 10 transactions.....	45
Figure III.2 : Treillis des ensembles fermés relatif à B.....	50
Figure III.3 : Extraction des ensembles fermés fréquents à l'aide du programme Close.....	51
Figure III.4 : Étapes de l'algorithme <i>Apriori</i> avec un support minimal de 30.....	56
Figure IV.1 : Diagramme de cas d'utilisation de système.....	66
Figure IV.2 : Diagramme de séquence de cas d'utilisation Extraction».....	67
Figure IV.3 :Diagramme de séquence de cas d'utilisation « Résultats ».....	68
Figure IV.4 :La forme principale.....	72
Figure IV.5 :La forme de traitement.....	73
Figure IV.6 :La forme de statistiques.....	74
Figure IV.7 :Interface des listes des produits (fréquentes ,fermées, maximal).....	75
Figure IV.8 :Interface de liste des règles.....	76

Introduction générale

Au vu du nombre croissant de grandes bases de données, déterminer la façon dont sont organisées les données, les interpréter et en extraire des informations utiles est un problème difficile et ouvert. À l'heure actuelle, notre capacité à collecter les données de tous types outrepassent nos possibilités d'analyse, de synthèse et d'extraction de connaissances dans les données [Han-06].

Une communauté s'est créée au milieu des années 90 autour de l'Extraction de Connaissances dans les Données (ECD) pour tenter d'apporter des solutions aux chercheurs et aux industriels. L'offre en matière d'outils d'ECD est en net accroissement, et certains de ces outils sont actuellement commercialisés comme par exemple *Purple Insight*.

Le domaine médical a connu une très grande utilisation des techniques de data mining et particulièrement, les données pharmaceutiques, où le pharmacien s'oppose à un grand problème, l'organisation des médicaments, l'approvisionnementetc.

Dans ce cadre d'étude, nous avons essayé de trouver une solution à ce problème en proposant l'application d'un algorithme d'extraction des règles à partir des données pharmaceutiques au niveau de la pharmacie.

Pour cela nous avons utilisé l'algorithme A-priori qui permet de trouver les règles d'associations des produits pharmaceutiques les plus pertinents.

Ce thème est structuré comme suit. Nous présentons dans première section une approche assez récente de fouille de données (data mining). La seconde section introduit le cadre de ce travail et présente plus en détails la recherche de règles d'association qui constitue le cœur de nos travaux. La troisième section de cet article présente l'algorithme qui détermine les règles d'association présentes dans une base de données transactionnelle, pour un seuil de support et un seuil de confiance. Directement liée à la troisième section, la

quatrième section qui est consacré à l'étude technique et la réalisation et l'implémentation du logiciel.

Nous concluons cet article en rappelant les résultats obtenus et en proposant quelques pistes d'utilisation de ce logiciel et Nous déterminons leur intérêt.

Chapitre

I

Data Mining

I. Data mining :

I. 1. Introduction :

Historiquement, le DataMining est très jeune. Le concept apparaît en 1989 sous un premier nom de KDD (Knowledge Discovery in Databases, en français ECD pour Extraction de Connaissances à partir des Données), avant qu'en 1991 apparaisse pour la première fois le terme de DataMining ou « minage / fouille des données ». Comme l'expliquent fort bien Michael Berry et Gordon Linoff, ce concept – tel qu'on l'entend aujourd'hui, et surtout tel qu'on l'applique dans les services marketing – est étroitement lié au concept du « one-to-one relationship ». C'est à dire la personnalisation des rapports entre l'entreprise et sa clientèle.

I.2. Définition générale:

Le data mining est un processus d'extraction de connaissances valides et exploitables à partir de grands volumes de données.

Il a vocation à être utilisé dans un environnement professionnel et se distingue de l'analyse de données et de la statistique par les points suivants :

- Contrairement à la méthode statistique, le data mining ne nécessite jamais que l'on établisse une hypothèse de départ qu'il s'agira de vérifier. Ce sont des données elles-mêmes que sont déduites les corrélations intéressantes, le logiciel n'étant là que pour les découvrir (le data mining se situe à la croisée des statistiques, de l'intelligence artificielle, des bases de données).
- Les connaissances extraites par le data mining ont vocation à être intégrées dans le schéma organisationnel de l'entreprise ou de l'entité considérée. Le data mining impose donc d'être capable d'utiliser de manière opérationnelle les résultats des analyses effectuées, souvent dans des délais très courts. Le processus d'analyse doit permettre à l'organisation une réactivité (très) importante.
- Les données traitées sont issues des systèmes de stockage en place dans l'organisation et sont ainsi hétérogènes, multiples, plus ou moins structurées. Leur raison d'être n'est donc *a priori* pas l'analyse (sauf dans le cas d'un entrepôt de données). Cela impose de disposer de systèmes performants de préparation ou de manipulation de données.

Le data mining se propose de transformer en information, ou en connaissance, de grands

volumes de données qui peuvent être stockées de manière diverse, dans des bases de données relationnelles, dans un (ou plusieurs) entrepôt de données (datawarehouse), mais

qui peuvent aussi être récupérées de sources riches plus ou moins structurées comme internet, ou encore en temps réel (appel à un call center, retrait d'argent dans undistributeur à billets...).

Lorsque la source n'est pas directement un entrepôt de données, il s'agit très souvent de construire une base de données ou un datamart dédié à l'analyse et aux analystes. Cela suppose d'avoir à sa disposition une palette d'outils de gestion de données (data management). On peut également structurer les données de l'entrepôt sous forme d'un hypercube OLAP, même si cela est assez rare en matière de data mining. [1]

Exemples d'applications:

On peut relever parmi les utilisations du data mining les exemples suivants :

- Analyser les comportements des consommateurs : ventes croisées (voir l'anecdote de Wal-Mart ci-après), similarités de comportements, cartes de fidélité, ...
- prédire la réponse à un mailing ou à une opération de marketing direct (par exemple pour en optimiser les coûts)
- Prédire l'attrition (ou churn) des clients : quels sont les indices de comportement permettant de détecter la probabilité qu'un client a de quitter son fournisseur (sa banque, son opérateur de téléphonie mobile, ...)
- Détecter des comportements anormaux ou frauduleux (transactions financières, escroquerie aux assurances, distribution d'énergie, ...)
- Rechercher des critères qui permettront d'établir ensuite une évaluation pour repérer les « bons » clients sans facteur de risque (Évaluation des risques-clients) et leur proposer peut-être une tarification adaptée (par exemple pour une banque ou une compagnie d'assurance).
- Suggérer lors d'un appel à un call center, en temps réel, une réponse de l'opérateur qui soit adaptée

Les outils de text mining (fouille de textes) associent en complément aux principes du data mining, l'analyse lexicographique et/ou linguistique multilingue des données non structurées comme les e-mails, les réponses à des questionnaires ou à des enquêtes, les réclamations clients, les blogs Internet, les news en ligne, les tchats... en tentant de déterminer par exemple le degré de satisfaction (ou d'insatisfaction) de la clientèle. Ses applications les plus rudimentaires sont la lutte contre le spam, ou pourriel, ainsi que

l'analyse de contenu. De plus ambitieuses concernent l'évaluation par analyse de dépêches de presse, de l'image d'une société, d'un climat politique ou boursier, ou la catégorisation automatique d'informations. [1]

I.3. Principe et spécificité:

I.3.1. Principe:

Le data mining est un processus d'analyse dont l'approche est différente de celle utilisée en statistique. Cette dernière présuppose en général que l'on se fixe une hypothèse que les données permettent ou non de confirmer. Au contraire, le data mining adopte une démarche sans a priori (approche pragmatique) et essaie ainsi de faire émerger, à partir des données brutes, des inférences que l'expérimentateur peut ne pas soupçonner, et dont il aura à valider la pertinence.

Le data mining tente alors de réaliser un arbitrage entre validité scientifique, interprétabilité des résultats et facilité d'utilisation, dans un environnement professionnel où le temps d'étude joue un rôle majeur et où les analystes ne sont pas toujours des statisticiens.

I.3.2. Le data mining est un processus:

Plus qu'une théorie normalisée, le data mining est un processus d'extraction de connaissances métiers comportant les étapes principales suivantes :

- Formaliser un problème que l'organisation cherche à résoudre en termes de données
- Accéder aux données appropriées quelles qu'elles soient
- Préparer ces données en vue des traitements et utilisations futurs
- Modéliser les données en leur appliquant des algorithmes d'analyse
- Évaluer et valider les connaissances ainsi extraites des analyses
- Déployer les analyses dans l'entreprise pour une utilisation effective

Ce processus est cyclique et permanent; le data mining rend dès lors plus compréhensible, "visible", l'activité de l'organisation, et permet de rationaliser le stockage de l'information et des données. Le data mining ne consiste pas en une succession d'études ad hoc mais a pour objectif de capitaliser des connaissances acquises sous forme de connaissances explicites.

Il conduit donc à mieux structurer les contenus nécessaires à l'ingénierie des connaissances. C'est sa principale raison d'être ; on peut comparer de façon lointaine cette activité à celle de conceptualisation au cours de l'apprentissage humain : « Une

bonne compréhension est intimement liée à une bonne compression » explique Donald Knuth[1] , l'une comme l'autre utilisant une connaissance de corrélations pour représenter - et donc manier - l'information sous forme plus concise.

I.3.3. Méthode :

Le data mining se propose d'utiliser un ensemble d'algorithmes (voir infra) issus de disciplines scientifiques diverses (statistiques, intelligence artificielle, base de données) pour construire des modèles à partir des données, c'est-à-dire trouver des schémas « intéressants » (des patterns ou motifs en français) selon des critères fixés au départ, et extraire de ces données un maximum de connaissances utiles à l'entreprise.

Voir par exemple les articles :

- Motif de conception
- algorithmes génétiques,
- Arbre de décision

I.3.4. Anecdotes :

Les premiers essais de fouilles de données se firent historiquement sur des dépouillements de millions de tickets de caisse de supermarché tels que mémorisés par les caisses enregistreuses.

À l'origine de la vulgarisation des méthodes et des algorithmes de la fouille de données (en l'occurrence, l'analyse des associations), il y aurait eu la mise en évidence par les magasins Wal-Mart d'une corrélation très forte entre l'achat de couches pour bébés et de bière le samedi après-midi[2] . Les analystes s'aperçurent alors qu'il s'agissait des messieurs envoyés au magasin par leur compagne pour acheter les volumineux paquets de couches pour bébé. Les rayons furent donc réorganisés pour présenter côte à côte les couches et les packs de bière ... dont les ventes grimpèrent en flèche ! Cette image plus ou moins véridique illustre le retour sur investissement (ROI) des travaux de data mining et plus généralement d'informatique décisionnelle.

D'autres analyses, en particulier d'indices précurseurs de modification des habitudes d'achat, se montrèrent d'un plus grand intérêt dans la pratique.

Un autre exemple souvent cité est le suivant : si on baisse le prix du Coca-Cola de 5%, on va par exemple en augmenter les ventes de 15%, ce que l'on savait sans data mining. Mais le data mining révèle l'élément inattendu (bien qu'évident a posteriori), à savoir que les ventes des cacahouètes vont augmenter dans une proportion voisine (sans doute suite à l'association d'idées : « Puisque j'achète du Coca, il me faut aussi des cacahouètes » ; le data mining ne fait pas d'hypothèse sur le sujet). Si la marge sur le Coca-Cola est

relativement faible, et celle sur les cacahuètes importante, la conclusion s'impose d'elle-même : baisser le prix du Coca-Cola est un moyen de vendre davantage de cacahouètes.

Par le côté quasi-déterministe qu'elle présente du comportement du consommateur, cette approche provoque souvent une réaction de scepticisme, mais dès lors que les résultats de sa mise en œuvre sont au rendez-vous, ce qui est quelquefois le cas, son adoption ne pose ensuite pas de problème particulier.

I.4. Algorithmes:

Résoudre une problématique avec un processus de data mining impose généralement l'utilisation d'un grand nombre de méthodes et algorithmes différents. On peut distinguer 3 grandes familles d'algorithmes :

I.4.1. Les méthodes non- supervisées:

Elles permettent de travailler sur un ensemble de données dans lequel aucune des données ou des variables à disposition n'a d'importance particulière par rapport aux autres, c'est-à-dire un ensemble de données dans lequel aucune variable n'est considérée individuellement comme la cible, l'objectif de l'analyse. On les utilise par exemple pour dégager d'un ensemble d'individus des groupes homogènes (typologie), pour construire des normes de comportements et donc des déviations par rapport à ces normes (détection de fraudes nouvelles ou inconnues à la carte bancaire, à l'assurance maladie...), pour réaliser de la compression d'informations (compression d'image)... Voici une liste non exhaustive des techniques disponibles :

- Techniques à base de Réseau de neurones : carte de Kohonen (SOM/TOM) (Carte Auto Adaptative)...
- Techniques utilisées classiquement dans le monde des statistiques : classification ascendante hiérarchique, k-means et les nuées dynamiques (Recherche des plus proches voisins), les classification mixtes (Birch...), les classifications relationnelles...
- Les techniques dites de recherche d'associations (elles sont à l'origine utilisées pour faire de l'analyse dite de panier d'achats ou de séquences, c'est-à-dire pour essayer de savoir parmi un ensemble d'achats effectués par un très grand nombre de clients et de produits possibles, quels sont les produits qui sont achetés simultanément (pour un supermarché par exemple ; elles sont également appliquées à des problèmes d'analyse de

parcours de navigation de site web). Ces techniques peuvent donc être utilisées de manière supervisées) : algorithmes a priori, GRI, Carma, méthode ARD...

- Analyses de liens

I.4.2. Les méthodes supervisées:

Leur raison d'être est d'expliquer et/ou de prévoir un ou plusieurs phénomènes observables et effectivement mesurés. Concrètement, elles vont s'intéresser à une ou plusieurs variables de la base de données définies comme étant les cibles de l'analyse. Par exemple, on utilisera ce type de méthode lorsque l'on cherchera à comprendre pourquoi un individu a acheté un produit plutôt qu'un autre, pourquoi un individu a répondu favorablement à une opération de marketing direct, pourquoi un individu a contracté une maladie particulière, pourquoi un individu a visité une page d'un site web de manière répétée, pourquoi la durée de vie après la contraction d'une maladie varie selon les malades... Voici une liste non exhaustive des techniques disponibles :

- Techniques à base d'arbres de décision (Arbre de décision) : CART, CHAID, ECHAID, QUEST, C5, C4.5, les forêts d'arbres...
- Techniques statistiques de régressions linéaires et non linéaires au sens large : Régression linéaire, Régression linéaire multiple, Régression logistique binaire ou multinomiale, Probit binaire, multinomial ou ordonné, Tobit, Cauchit, modèle Gamma, binomial négatif, log-log, Analyse discriminante linéaire ou quadratique, régression de cox, modèle linéaire généralisé, régression PLS, régressions non paramétrique, équations structurelles ... (voir aussi économétrie)
- Techniques à base de Réseau de neurones : perceptron mono ou multicouches avec ou sans rétropropagation des erreurs, les réseaux à fonction radiale de base...
- Techniques à base d'algorithme génétique.
- Techniques à base d'Inférence bayésienne (Réseau bayésien).
- Le Raisonnement par cas
- Le Filtrage collaboratif

I.4.3. Les méthodes de réduction de données:

Elles permettent de réduire un ensemble de données volumineux à un ensemble de taille plus réduite, épuré de ce que l'on considèrera comme de l'information non pertinente ou non signifiante, comme du bruit. Elles sont ainsi très souvent, mais pas systématiquement, utilisées en amont des techniques supervisées ou non supervisées. Elles sont notamment très complémentaires des techniques non supervisées classiquement utilisées dans le domaine de la statistique.

- Techniques d'Analyse factorielle : Analyse en composantes principales, Analyse en composantes indépendantes, analyse factorielle des correspondances, analyse des correspondances multiples, analyses factorielles (maximum de vraisemblance, moindres carrés non pondérés, avec ou sans rotation orthogonale ou oblique)...
- Techniques de positionnement : positionnement multidimensionnel...

I.4.4. Pourquoi tant d'algorithmes ?

Parce que nous venons de voir qu'ils n'ont pas tous le même objet, parce qu'aucun n'est optimal dans tous les cas, parce qu'ils s'avèrent en pratique complémentaires les uns des autres et parce qu'en les combinant intelligemment (en construisant ce que l'on appelle des modèles de modèles ou métamodèles) il est possible d'obtenir des gains de performance très significatifs, si l'on prend bien garde d'éviter des problèmes de sur-ajustement des modèles ainsi obtenus (voir à ce sujet l'article Arbre de décision et son paragraphe traitant du problème de sur-ajustement des modèles). Encore faut-il être en mesure de réaliser ces combinaisons facilement, ce que permettent les logiciels ateliers de Data Mining, par opposition aux outils de statistiques classiques dans lesquels l'opération est beaucoup plus délicate en pratique.

I.5.le Data mining et statistique:

Généralement, une approche statistique consiste à formuler des hypothèses qui sont confirmées ou infirmées à l'aide de tests statistiques .Si l'étude est correctement menée , on formule les hypothèses , on en déduit un plan d'expérience, on recueille les données , puis analyses .

En data mining , la recherche d'information est moins guidée . Cependant, le data mining repose sur un ensemble des méthodes issues des statistiques et de l'intelligence artificielle. [Tuf-07]

I.5.1.Quelques repères :

- Quelques centaines d'individus
- Quelques variables recueillies avec un protocole spécial(échantillonnage, plan d'expérience.....)
- Fortes hypothèses sur les lois statistiques suivies
- Les modèles sont issus de la théorie et confrontés aux données
- Méthodes probabilistes et statistiques
- Utilisation en laboratoire

I.5.2. Analyse des données : [Tuf-07]

- Méthodes essentiellement descriptives
- Quelques dizaines de milliers d'individus
- Quelques dizaines de calculs et de représentation visuelle

I.5.3. data mining : [Tuf-07]

- Plusieurs millions d'individus
- Plusieurs centaines de variables
- Nombreuses variables non numériques , parfois textuelles
- Données recueillies avant l'étude , et souvent à d'autres fins
- Données imparfaites , avec des erreurs de saisie , de codification, des valeurs manquantes , aberrantes
- Population constamment évolutive (difficulté d'échantillonner)
- Nécessité de calculs rapides , parfois en temps réel
- On ne recherche pas toujours l'optimum mathématique , mais le modèle le plus facile à appréhender par des utilisateurs non statisticiens
- Faibles hypothèses sur les lois statistiques suivies
- Les modèles sont issus des données et on tire des éléments théoriques
- Méthodes statistiques , d'intelligence artificielle et de théorie de l'apprentissage ("machine Learning")
- Utilisation en entreprise

I.6. Etapes de création d'un projet data mining:

- Choix du sujet _ définition des objectifs
- Inventaire des données existantes
- Collectes , nettoyage et mise en forme des données
- Etudes statistiques de base d'analyse
- Mise en œuvre des algorithmes classification , scoring)
- Elaboration des modèles
- Validation et choix d'un modèle
- Déploiement du modèle
- Formation des utilisateurs
- Suivi des modèles

I.7. Processus Data Mining:

le Data Mining est un processus non-trivial d'identification de structures inconnues, valides et potentiellement exploitables dans les bases de données. [Fay-96]

Plusieurs intervenants industriels ont proposé une formalisation de ce processus, sous la forme d'un guide méthodologique nommé CRISP-DM pour Cross Industry Standard Process for Data Mining .

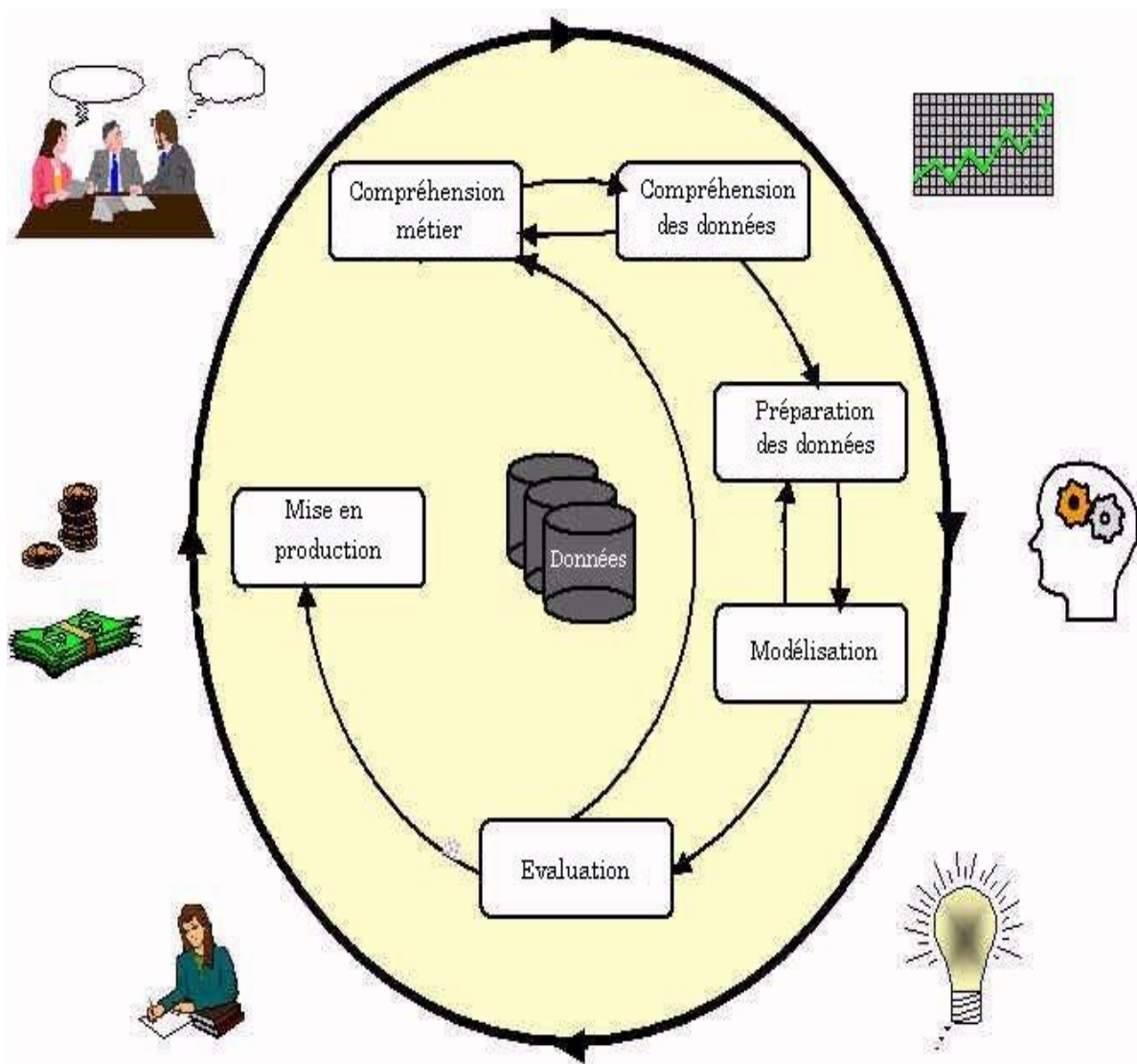


Figure I.1 : Processus Data Mining CRISP-DM.

Le modèle CRISP-DM, illustré sur la figure 1.1, propose de découper tout processus Data Mining en 6 phases :

1. La phase de recueil des besoins (de l'anglais business understanding), fixe les objectifs industriels et les critères de succès, évalue les ressources, les contraintes et les hypothèses nécessaires à la réalisation des objectifs, traduit les objectifs et critères industriels en objectifs et critères techniques, et décrit un plan de résolution afin d'atteindre les objectifs techniques.
2. La phase de compréhension des données (de l'anglais data understanding), réalise la collecte initiale des données, en produit une description, étudie éventuellement quelques hypothèses à l'aide de visualisations et vérifie le niveau de qualité des données.
3. La phase de préparation des données, consiste en la construction d'une table de données pour modélisation. Nous nous y intéressons plus particulièrement par la suite.
4. La phase de modélisation, procède à la sélection de techniques de modélisation, met en place un protocole de test de la qualité des modèles obtenus, construit les modèles et les évalue selon le protocole de test.
5. La phase d'évaluation estime si les objectifs industriels ont été atteints, s'assure que le processus a bien suivi le déroulement escompté et détermine la phase suivante : retour en arrière ou déploiement.
6. La phase de déploiement industrialise l'utilisation du modèle en situation opérationnelle, définit un plan de contrôle et de maintenance, produit un rapport final et effectue une revue de projet .

Le modèle CRISP-DM est essentiellement un guide méthodologique pour la conduite d'un projet Data Mining. Les phases initiales et finales s'apparentent à des activités d'expertise en organisation, consulting, bases de données et développement informatique. Elles supposent une implication humaine importante. Seules les phases centrales sont partiellement automatisables, notamment, la phase de modélisation, naturellement consommatrice en techniques de modélisation statistique.

I.8.Synthèse des besoins :

Que fait le DataMining ? On peut lister cinq grandes fonctionnalités qui sont la base de tout ce que l'on peut essayer de faire quand on veut « miner » ses données :

- description
- classification
- regroupement par similitudes
- estimation

- prédiction

Ces 5 fonctions se regroupent en deux familles : exploration et confirmation. On peut illustrer cette « filiation » sur le schéma ci-dessous.

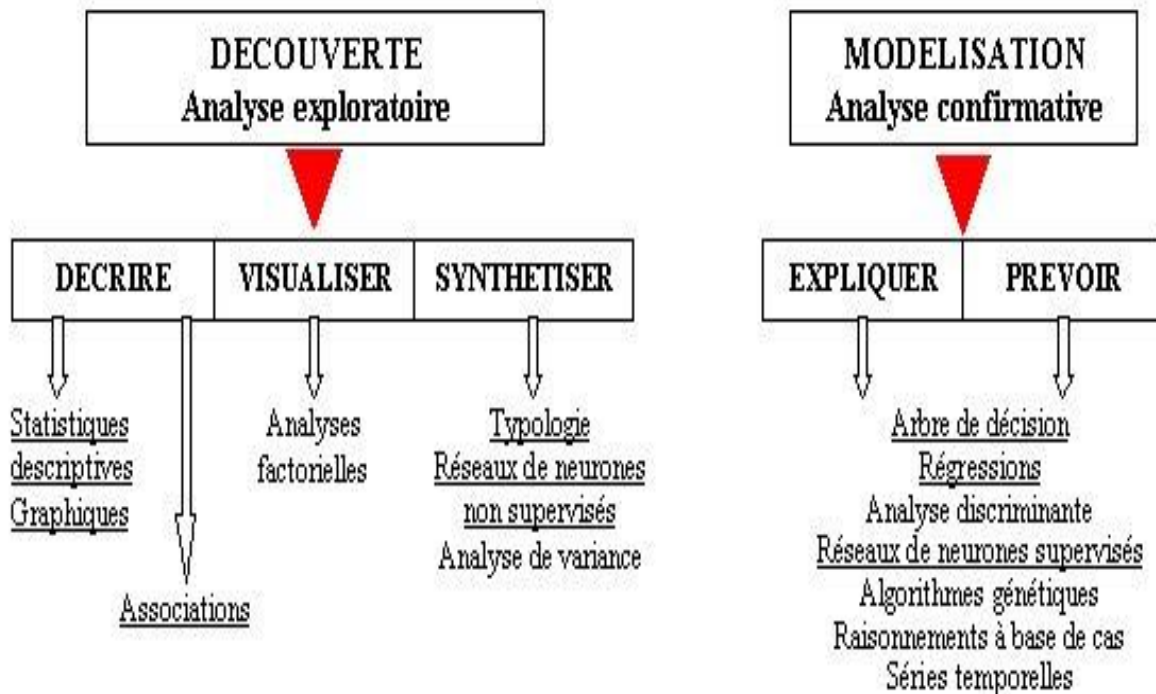


Figure I.2 : le schéma du grandes fonctionnalités de « miner » les données

Sur ce schéma, les fonctionnalités directement accessibles depuis Sas Enterprise Miner (version 3) sont soulignées. Un signe accompagne les fonctionnalités apparues avec la version 4.1. Certaines autres sont implémentables manuellement au prix d'une bonne connaissance de la programmation en SAS (on perd alors une bonne partie de l'avantage conféré par l'interface graphique). Il s'agit des techniques factorielles (l'analyse des données dite « à la française »), de l'analyse de la variance, de l'analyse discriminante et des séries temporelles (modèles et prévision). Les algorithmes génétiques, qui souffrent de ne revêtir que des formes trop spécialisées, et le raisonnement à base de cas (ou de mémoire) sont absents du logiciel et ne se mettent pas en œuvre dans l'univers SAS sans une bonne dose de patience, d'efforts et de réflexion algorithmique.

Sur l'ensemble du schéma, nombre des tâches, à cause des techniques qu'elles requièrent pour être menées à bien, sont rattachées à des domaines plus ou moins pointus de la

statistique. On peut donc également lister les points de théorie statistique qu'il faudra que les utilisateurs connaissent pour faire fonctionner efficacement SAS Entreprise Miner.

- statistiques descriptives (khi-2, boxplot, adéquation à une loi – QQ plots en particulier – , tests)
- classification (principe de la classification ascendante hiérarchique)
- cartes auto-organisées (type Kohonen)
- régressions (logistique et moindres carrés ordinaires)
- réseaux de neurones (principes généraux et différences entre perceptron et RBF)
- arbres de segmentation (principe général et différences entre Chaid, CART et C4.5)
- techniques de score

I.8.1.Synthèse des problématiques associées au DataMining:

Connaître les techniques que l'on peut mettre en place ne permet pas toujours de savoir comment et dans quel cadre les utiliser. En particulier, dans la masse d'exemples proposés ci-dessus, il convient de distinguer quelles sont les grandes problématiques. Celles que l'on peut à bon droit associer à une approche de DataMining, peuvent être regroupées en quelques catégories synthétiques :

- Action commerciale ou marketing sur des **populations très ciblées** (ce que l'on nomme « niches marketing ») : on fait alors appel aux techniques de segmentation pour chercher et caractériser ces microcosmes.
- **Etude du « churn »** ou perte de clients : on cherche à identifier le profil de clients risquant de partir. Pour cela, on use de typologies. On peut aussi construire un score après une modélisation sur la variable parti/resté (sur des données historisées). Une fois identifiés, les clients « à risques » seront relancés afin de mieux cerner leurs griefs et y proposer des solutions.
- **Tarifications ou services adaptés** à des profils de clients particuliers (sans pour autant se fixer sur des micro-populations particulièrement rentables) : une typologie des clients permet, en ne l'affinant pas outre mesure comme dans la recherche de niches, de dégager des grandes « familles » dans la population cliente, et de proposer à son catalogue des offres adaptées sans être trop spécifiques d'une frange très étroite de la clientèle.
- Enfin, reste à part les **problèmes d'associations** : aussi bien l'analyse du panier de la ménagère (étude des tickets de caisse) que les groupements de services sont des problématiques spécifiques, qui ne sont que rarement intégrées aux autres traitements du DataMining.

Qu'est-ce que le DataMining ? Pour en faire la synthèse, on est obligé de répondre à cette question de façon lapidaire et partielle, tant elle fait l'objet de nombreux ouvrages et débats. Il consiste à découvrir des enseignements utilisables et rentables pour l'entreprise dans une grosse masse de données.

Les différentes techniques mises en œuvre se retrouvent dans SEM, elles n'ont pas toutes la même complexité. Enfin, les domaines d'applications sont très vastes. [1]

I.9. Quels logiciels utilise t'on en data mining ?

1. Les logiciels usuels de statistique : SAS, R , Splus ,
2. Des logiciels de calcul numérique : matlab
3. Des parties de logiciels dédiées : SAS/EM , SPSS/Clémentine , Statsoft/Statistica Data Miner ,Insight/ Insightful Miner SPAD
4. Des logiciel libres : weka , tanagra

I.10. Les besoins et les attentes des utilisateurs

Nous avons préféré partir des situations rencontrées dans des entreprises réels. Que ce soit par des entretiens avec des chargés d'études et des responsables de « services clients», ou par les cas concrets proposés à l'appui d'une démonstration dans la littérature (comme c'est souvent le cas dans Berry et Linoff), la description du quotidien du DataMining sera faite surtout par l'exemple. Nous allons commencer par une situation qui résume la quasi-généralité des préoccupations rencontrées. Ensuite, nous les détaillerons en voyant d'autres exemples avant de chercher à montrer la diversité des approches rencontrées.

I.10.1. Un exemple paradigmatique

Une responsable des études commerciales de La Redoute définit ainsi les trois attentes de son service marketing...

En premier lieu, structurer et agréger les données : la base clients du vériciste comporte quelque 16 millions d'adresses, assorties de divers renseignements sur chaque client. Trier, recouper et organiser cette masse d'information est effectivement la priorité et le préalable à toute étude sérieuse.

Deuxième point, transformer cette somme de renseignements en connaissances. La distinction entre ces deux termes, « renseignements » et « connaissances » est la clé du DataMining et sa raison d'exister.

Les renseignements sont des données qui existent dans les bases de l'entreprise. Elles y

« dorment » et ne servent à personne. Ce fut longtemps le cas des données clientèle, puisque, focalisée sur les produits, l'attention du service clients ne se tournait pas vers les acheteurs. Les connaissances, en revanche, sont utiles à l'entreprise. Elles aident à la prise de décision. Elles entrent en ligne de compte dans les modèles. Elles permettent de faire des choix et d'orienter la politique commerciale de la firme.

Depuis longtemps, les entreprises achètent à l'INSEE les données du recensement qui ne servaient qu'à juger du taux de pénétration d'un produit sur une aire géographique donnée. On ne corrigeait guère ces taux des profils d'acheteurs spécifiques à chaque région (rurale, urbanisée, frontalière, intérieure, riche ou pauvre, ...) et on appliquait peu ou prou le même calcul à Paris et à Mende. Ne sortant pas de ce domaine qui touche plus les tableaux de bords que les problèmes décisionnels, ces données restaient à l'état de renseignements. Ce n'est que récemment qu'on les a prises en compte. Elles servent aujourd'hui à mieux connaître les acheteurs que l'on est susceptible de rencontrer dans telle ou telle zone de France. Elles sont devenues des connaissances.

Au sein de cette démarche en particulier, la transmutation de l'information en connaissance se traduit par un élagage des données redondantes autant que par la création d'indicateurs synthétiques pour résumer les chiffres à interpréter.

Troisième et dernière préoccupation du service clients de La Redoute, hiérarchiser la clientèle selon des critères de rentabilité. Cette dernière étape pourra sembler pour un statisticien « classique » comme la seule digne d'intérêt. En effet, les Data Warehouses, par les volumes mis en jeu, ont décuplé l'acuité du nettoyage des données avant emploi ; cela n'a pas toujours été le cas. De la même façon, auparavant on avait si peu de données à faire ingérer au modèle qu'on ne distinguait pas connaissances et renseignements. La troisième étape du DataMining chez le vériciste est donc la plus classique de toute. Elle consiste à utiliser sur les données l'arsenal des techniques statistiques à notre disposition pour les faire parler. On distingue, dans ces techniques, un DataMining descriptif, différent du DataMining explicatif. Le premier cherche à cerner les comportements, et aide à légitimer l'étape précédente où l'on souhaitait décrire avec le moins de chiffres possibles la réalité. Le second vise à expliquer lesdits comportements, et à en inférer une conduite à tenir.

On arrive ainsi à une prise de décision, qui sera suivie d'effets. Ces conséquences, conformément à la méthodologie proposée plus haut, seront quantifiées (logiciels de suivi des actions marketing) en deux étapes : d'abord un test sur un échantillon représentatif avant de lancer, si les résultats sont concluants, la même campagne en grandeur réelle.

Les enseignements tirés de cette démarche seront utilisés pour la définition de la suivante, ce qui nous rappelle que la marche à suivre est cyclique.

I.10.2. La prédiction, premier axe de recherche:

L'utilisation massive du DataMining est centrée sur des problèmes décisionnels. C'est à dire sur la prédiction (avec un niveau de probabilité) d'une variable polytomique, ordonnée ou non (elle est souvent dichotomique : accorder un crédit / le refuser, le client répondra au mailing / ne répondra pas, etc...). Une fois ces valeurs prédites avec leur probabilité, on peut construire un score, c'est à dire une échelle continue de valeurs qui note au mieux les individus les plus « intéressants » (il faut alors construire un modèle de coûts ou de profits).

On trouve des exemples de cette utilisation dans tous les secteurs où le marketing décisionnel est développé, ainsi la banque, la finance, l'assurance, la vente par correspondance, la grande distribution, ...

Modéliser sert aussi au cercle vertueux du DataMining, celui qui est explicité dans le processus CRISP-DM. La modélisation permet en effet de tirer les enseignements d'une campagne précédente, en étudiant quels ont été les clients sensibles à une promotion. Quand c'est réalisable, il est bon de comparer le score *a priori*, sur la foi duquel on a ciblé cette campagne promotionnelle, et le score *a posteriori* qui est représentatif des résultats obtenus. On peut ainsi se servir du second pour corriger le premier, déterminer ses forces et ses faiblesses. Pour cela, il suffit de modéliser l'écart entre les deux, avec une matrice de coûts idoine. Les explicatives seront les mêmes que dans la construction des deux scores : leur contribution au modèle rend ainsi compte de leur part dans la réussite ou l'échec du score *a priori*. Elle permet aussi d'éliminer ou de se concentrer sur certaines caractéristiques en vue du score à produire pour la prochaine campagne.

Autre exemple de « minage des données » à but décisionnel : l'appétence à posséder un type particulier de carte bleue, pour les clients d'un groupe bancaire. Quels facteurs font que les clients vont se rendre acquéreurs de ses produits ou, au contraire, vont les abandonner ? Il y a ici une valeur descriptive au processus de modélisation. Mais, par ailleurs, un modèle plus abscons, une « boîte noire » qui rendrait mieux compte du comportement de la clientèle (en mettant en jeu des effets non linéaires), est également le bienvenu. Il permet d'affubler d'un score les futurs acquéreurs, et de détecter ceux qui vont résilier leur contrat parmi les clients déjà possesseurs du produit.

I.10.3. Second axe, la création de typologies:

Une demande à peine moins courante est de construire des typologies. Il s'agit d'exploiter les méthodes dites « non supervisées » que regroupe le DataMining, comme les cartes de Kohonen ou les méthodes des nuées dynamiques. Une typologie permet, sans idée d'expliquer a priori UNE variable et une seule, de distinguer des classes d'individus aux mœurs différentes au sein de sa clientèle.

Il s'agit par exemple de la demande du service marketing d'EDF, souhaitant mieux connaître sa clientèle ; on peut aussi citer la Redoute, dont une des préoccupations est de créer des classes de clients dont il faut s'occuper différemment. On peut avoir sa clientèle composée de médecins, comme c'est le cas dans le cadre de l'industrie pharmaceutique. Dans ce cas, la typologie s'effectue sur le type de prescriptions qu'ils ont coutume de faire, pour leur proposer la gamme de médicaments *ad hoc*.

Cette analyse peut aussi servir aux administrations (Ministère de la Santé, Caisse Nationale d'Assurance Maladie) et aux organismes parapubliques qui font des études dans le domaine de santé et des médicaments (CREDES³ par exemple).

Les mêmes peuvent encore se pencher sur une typologie des hôpitaux et cliniques afin de légitimer (ou d'infirmer) les conclusions des enquêtes-choc d'associations de défense des consommateurs, lesquelles, tous les ans, nous proposent la liste des établissements de santé à éviter. Une typologie permettrait de cerner les classes où, effectivement, des services sont déficients, et de donner des arguments valables car mathématiques et statistiques pour alimenter les débats.

La grande distribution doit résoudre des problèmes descriptifs d'un type particulier : c'est ce que l'on appelle « l'analyse du panier de la ménagère ». L'exemple le plus parlant consiste à étudier une masse de tickets de caisse. D'un client à l'autre, quelles sont les combinaisons qui reviennent le plus souvent – il s'agit du problème des « règles d'associations » ? Lors d'achats successifs, quels sont les enchaînements d'achats que l'on peut retrouver chez plusieurs clients – ce sont ce qu'on appelle des « séquences fréquentes » ?

Cette étude nécessite des aménagements particuliers dans les données, et n'est pas forcément la plus simple à mettre en place. En effet, il faut posséder, en sus de la « base client » nécessaire au DataMining, un fichier qui compte autant de lignes par client qu'il a acheté d'items. Il faut donc pouvoir accéder également à un fichier non agrégé.

Mais une étude brute, particulièrement dans le cas des tickets de caisse, offre rarement des résultats exploitables : la faute en incombe à la représentation des données. Chaque

produit qui passe en caisse est en effet repéré par son code-barre, qui représente une clé à 14 chiffres appelée Gencode. On a ainsi, dans la base, une grande précision sur l'item acheté : marque, produit exact, conditionnement... Il existe une aussi grande différence entre deux bouteilles de vin blanc de mêmes cépage et millésime et de prix voisins, mais de producteurs différents, qu'entre un kilo de pommes et une paire de chaussettes. Tous ont, en effet, des Gencodes différents. De surcroît, pour ne rien arranger, le « radical » d'un Gencode représente le code du fournisseur du produit : il peut donc s'agir d'une grande variété d'items différents. Par exemple, un Gencode commençant par le code de Coca-Cola France peut être celui d'une boisson gazeuse (Coca-Cola, Sprite) ou d'un jus de fruits (Minute Maid)... Sans parler des grands groupes de l'agro-alimentaire comme Nestlé ! Il faut donc définir ce que l'on appelle des taxinomies, c'est à dire des tables de regroupement des produits par familles. On agrège ainsi les Gencodes par items de même type (tous les vins de table, toutes les pièces de porc, toutes les confitures), pour obtenir divers niveaux de finesse dans l'analyse. On peut ainsi vérifier très rapidement certains lieux communs du marketing, avec des regroupements très grossiers : viande, poisson, fruits, légumes, etc... Il faut ensuite trouver un niveau d'analyse qui apporte des enseignements intéressants, sans tomber dans la finesse excessive qui propose comme phénomène le plus courant l'association de deux Gencodes qui ne se sont trouvés ensemble qu'une dizaine de fois sur un million !

Par ailleurs, il peut être intéressant d'ajouter dans la base d'étude un certain nombre de paramètres pouvant influencer sur le type d'achats : région d'achat, taille du magasin (super, hypermarché, moyenne surface, ...), produit soldé, etc... On appelle ces paramètres des items virtuels, et ils sont mis au même niveau que des items « normaux ».

Le recours aux items virtuels permet d'étendre ce principe d'analyse au-delà de la grande distribution. Une banque pourra par exemple définir les types d'utilisation de ses cartes bleues par cette méthode. Pour un client, on énumérera ses retraits, demandes de RIB, éditions de positions de comptes, transferts de fonds, tous réalisés à des automates avec leur carte. L'ajout d'items virtuels comme le quantième du mois ou l'éloignement géographique de l'agence où est ouvert le compte du client permet alors d'analyser plus finement les comportements. On peut enfin compter au nombre de ces items virtuels les renseignements habituels des analyses descriptives sur un client : CSP, ancienneté dans la banque, et autres variables socio-démographiques. [1]

I.11.Conclusion :

Nous avons ici présenter quels sont les principaux avantages du data mining, et ainsi mieux cerner l'utilité de cette méthode.

- Le data mining aide à la prise de décision des dirigeants. Par l'analyse des données, la méthode peut « résumer » la situation et alors accélérer la prise de décision des dirigeants à un problème donné. Par contre, le data mining ne remplace pas ces dirigeants.
- Le data mining permet de faire des liens pertinents entre des données qui, à première vue, n'ont aucune corrélation.
- Cette méthode peut améliorer la satisfaction des clients en analysant leur besoins et en proposant des améliorations en fonction des événements passés.
- Permet d'effectuer des profils « type » : des profils de clients comme d'employés en fonction d'un test, pour prévoir l'évolution de ceux-ci aux

cotés de l'entreprise.

- Le data mining facilite le développement de nouveaux produits
- Accélère la gestion des stocks, des inventaires, de la logistique
- Peut augmenter les revenus tout en diminuant les coûts.

C'est évident, le data mining a été étudié pour augmenter et optimiser le rendement d'une entreprise ou l'amélioration d'un critère.

Dans le chapitre suivant nous allons décrire en détail une des tâches fondamentale de data mining, qui est l'extraction des règles d'associations.

Chapitre

II

Règles d'associations

II. Règles D'association:

II.1. Introduction :

L'extraction des règles d'association est devenue aujourd'hui l'une des tâches les plus populaires de la fouille de données, et ce, depuis les travaux de Agrawal et al. [AIS93, AS94]. L'analyse de panier de ménagère est l'une des applications typiques de l'extraction des règles d'association. Elle a pour but de dégager les relations intelligibles entre les attributs dans une base de données. Dans le cas de l'analyse du panier de ménagère, l'extraction des règles d'association permet d'analyser les tickets de caisse des clients particuliers afin de comprendre leurs habitudes de consommation, agencer les rayons du magasin, organiser les promotions, gérer les stocks etc. dans le naturel but d'améliorer le profit. Dans la base de données de vente, une transaction consiste à un ensemble d'articles achetés par un client particulier, appelés items ou attributs. Ainsi, une base de données est un ensemble de transactions qu'on appelle aussi base transactionnelle. Dans un tel contexte, une règle d'association est une implication conditionnelle entre des ensembles d'attributs dans une base transactionnelle. En d'autres termes, étant donné un ensemble d'attributs, le but de l'extraction des règles d'association est de découvrir "si l'occurrence d'un tel ensemble dans une transaction est associée à l'occurrence d'un autre ensemble d'attributs". Par exemple, "70% des clients achetant du lait et du thé achètent aussi du pain" est une règle d'association associant les attributs lait et thé à l'attribut pain. Ce problème est loin d'être trivial, vu le nombre exponentiel du nombre d'attributs de la base transactionnelle. Par ailleurs, une base transactionnelle stocke des millions de transactions sur des milliers d'attributs

Ce chapitre concerne les processus d'extraction des règles d'association d'un contexte de la fouille de données.

II. 1. Définition d'une règle d'association :

Une règle d'association est une relation d'occurrence qui lie deux ou plusieurs items particuliers. Ceci s'illustre plus facilement à l'aide d'un exemple. Si le contenu des paniers d'achats révèle qu'un grand nombre de clients qui achètent du pain, achètent également du beurre, cela signifie qu'il existe une relation d'occurrence simultanée forte entre ces deux items. Cette relation est appelée association et de celle-ci s'induit la règle d'association suivante : pain => beurre. Cette règle d'association est dite classique ou

traditionnelle, puisque les deux événements, l'achat du pain et l'achat du beurre, se déroulent simultanément

lors de la même transaction. Lorsque l'occurrence des items est étalée selon un ordre spécifique, par exemple les clients qui achètent du pain, achètent également du beurre dans les jours qui suivent, les règles d'associations entre ces items ordonnés sont qualifiées d'associations séquentielles. L'objectif des règles d'associations, traditionnelles ou séquentielles, consiste à découvrir les associations fréquentes dans les données étudiées. [Bla-05]

II. 2 Concepts et définition:

Les règles d'associations se définissent comme un ensemble de techniques recherchant des relations intéressantes parmi les items contenus dans une série de données particulière [Han-06] . Ces règles, telles qu'on les connaît actuellement en analyse de données, ont été introduites et façonnées par deux articles publiés par Agrawal *et al.* en 1993 et 1994. Dans le premier, les auteurs ont défini le problème et les concepts entourant les règles d'associations, tandis que dans le second, ils ont introduit l'algorithme et l'heuristique nommé *Apriori*.

Depuis, le nombre de travaux consacrés aux règles d'associations ne cesse d'augmenter et l'algorithme *Apriori* demeure toujours un algorithme central qui se retrouve dans la plupart des algorithmes consacrés aux règles d'associations plus complexes.

Les règles d'associations classiques sont également connues sous le nom de *market basket analysis* (analyse du panier d'achat) puisque les premiers travaux ont été réalisés à partir de données provenant de supermarchés. Ces travaux ont utilisé des règles d'associations afin d'identifier les items ou les groupes d'items achetés ensembles par un client lors d'une même transaction. De manière générale, une règle d'association s'énonce comme suit:

$$\text{pain} \Rightarrow \text{beurre} \text{ [support = 2\%, confiance = 60 \%]}. \text{ (2.1)}$$

La règle d'association de l'énoncé 2.1 se lit de la manière suivante: 60 % des clients qui ont acheté du pain ont également acheté du beurre. De plus 2 % des transactions totales enregistrées dans la base de données respectent cette règle. De façon générale, la confiance se définit comme la probabilité d'obtenir dans une transaction l'item résultant de la règle, dans ce cas-ci du beurre, sachant qu'un ou que d'autres items, dans ce cas-ci

le pain, s'y retrouvent également. Le support, quant à lui, est donné par la proportion des transactions de la base de données transactionnelles qui contiennent tous les items présents dans la règle. Ce petit exemple démontre l'avantage principal des règles d'associations: leur simplicité. Contrairement à d'autres méthodes d'analyse de données, les règles d'associations fournissent des réponses simples et facilement interprétables quel que soit le degré de complexité de la règle.

II.3 Utilité des règles d'associations:

Les règles d'associations sont appliquées dans plusieurs domaines. En marketing, par exemple, elles permettent d'identifier les produits ou services qui sont achetés lors d'une même transaction ou par un même client dans le temps et offrent donc la possibilité d'identifier des opportunités de ventes croisées. En analysant l'ordre dans lequel les internautes accèdent aux pages d'un site WEB, les règles d'associations séquentielles permettent d'entrevoir quelles modifications rendraient le site plus convivial, permettant ainsi aux internautes de trouver rapidement les informations recherchées. Les règles d'associations séquentielles sont également utiles dans l'étude des comportements d'achats des consommateurs, car elles permettent de mettre en lumière les comportements d'achats à travers le temps. Par exemple, si les associations séquentielles indiquent que les propriétaires d'une marque et d'un modèle précis de voiture changent leur véhicule aux 18 mois, le concessionnaire pourra fidéliser sa clientèle en faisant parvenir un dépliant sur les nouveaux modèles disponibles quelques semaines avant la date du changement de voiture.

II.4 Signification d'une règle d'association:

La signification statistique des règles d'associations peut porter à confusion. L'origine de cette confusion provient sans doute de la manière dont les règles d'associations sont énoncées. [Han-06]

L'exemple discuté à la section précédente, pain => beurre, avec par exemple une confiance de 60 %, peut laisser croire qu'un client qui achètera du pain a 60 % de chances d'acheter également du beurre. Cette interprétation est toutefois erronée. L'interprétation de cette règle se fait plutôt de la manière suivante : dans la base de données étudiée, 60% des clients qui ont acheté du pain ont également acheté du beurre.

Les règles d'associations ne peuvent donc pas être utilisées comme outil de prédiction, du moins tant et aussi longtemps qu'elles n'ont pas été modifiées. Afin de rendre prédictive les règles d'associations, Freitas (2000) suggère deux modifications. Premièrement, les règles d'associations devront être enrichies de procédures qui empêchent le sur-ajustement (*overfitting*) et le sous-ajustement (*underfitting*) des données.

Deuxièmement, les règles devront être soumises à un ensemble de données «vierges », sur lesquelles les règles n'ont pas été entraînées et qui confirmerait la validité de ces dernières.

Liu et *al.* (1998) se sont penchés sur la question et proposent un classificateur basé sur les associations (CBA). [Liu-99] Ce problème n'est toutefois pas discuté d'avantage dans ce mémoire

II.5 Actionnabilité des règles d'association:

II.5.1 Terminologie et notations

Considérons un ensemble T de n transactions décrites par un ensemble I de variables qualitatives ou quantitatives. T est stocké sous forme de table dans une base de données relationnelle.

L'ensemble $I = \{i_1, i_2, \dots, i_m\}$ représente un ensemble de m attributs décrivant les n transactions de la base de données. Ces variables sont appelées des items dans la terminologie des règles d'association. L'ensemble $T = \{t_1, t_2, \dots, t_n\}$ représente un ensemble de transactions, une transaction étant un sous ensemble de I . Une conjonction d'items $Z = \{i_1, i_2, \dots, i_k\}$ non vide de I est appelé un itemset. Le nombre d'items k de l'ensemble Z constitue sa longueur. Un itemset de longueur k est nommé k -itemset.

Un ensemble d'items est dit **fréquent** si et seulement si il correspond à un motif fréquent dans la base de transactions. Nous définissons le support d'un motif comme étant le nombre de transactions dans T contenant ce motif divisé par le cardinal de T , nommé $\text{card}(T)$. Un seuil minimal de support minSup est fixé à partir duquel un ensemble d'items est dit fréquent.

II.5.2 Règles d'association

Une règle d'association est une liaison orientée entre variables booléennes de la forme $X \rightarrow Y$, où X et Y sont des itemsets ou négations d'itemsets qui n'ont pas d'items en commun. Elle traduit la tendance de Y à être vrai quand X est vrai, et peut s'énoncer de la manière suivante : « si un individu vérifie X alors il vérifie sûrement Y ». X est appelé

prémisse et Y conclusion de la règle. Les exemples d'une règle sont les individus qui vérifient la prémisse et la conclusion. En revanche, les contre-exemples sont les individus qui vérifient la prémisse et non la conclusion. À partir de deux variables X et Y, il est possible de construire huit règles différentes (cf. tableau II.1). [Bla-05]

1 : $X \rightarrow Y$	2 : $X \rightarrow \bar{Y}$	3 : $\bar{X} \rightarrow Y$	4 : $\bar{X} \rightarrow \bar{Y}$
5 : $Y \rightarrow X$	6 : $Y \rightarrow \bar{X}$	7 : $\bar{Y} \rightarrow X$	8 : $\bar{Y} \rightarrow \bar{X}$

TAB II.1 : Règles d'association générées à partir de deux variables X et Y

De plus, pour une règle $X \rightarrow Y$, nous utilisons les notations suivantes :

- $X \rightarrow Y$ est la règle contraire, $Y \rightarrow X$ la réciproque et $\bar{Y} \rightarrow \bar{X}$ la contraposée ;
- $n = |T|$, le nombre de transactions ou enregistrements ;
- $n_x = |X|$, le nombre de transactions satisfaisant X ;
- $n_y = |Y|$, le nombre de transactions satisfaisant Y ;
- $n_{xy} = |X \cap Y|$, le nombre de transactions satisfaisant à la fois X et Y ;
- $n_{\bar{x}y} = |(\bar{X} \cap Y)|$, le nombre de transactions satisfaisant X mais pas Y.

Soit x une variable booléenne qui est un itemset ou une négation d'itemset. La variable $\bar{x}$ est la négation de x. Nous notons X l'ensemble des transactions de T vérifiant x et n_x le cardinal de X. Le complémentaire de X dans T est $\bar{X}$ de cardinal $n - n_x$. La probabilité de l'événement x est vrai est notée $P(x)$. Elle est estimée par la fréquence empirique (estimateur du maximum de vraisemblance) : $P(x) = n_x / n$.

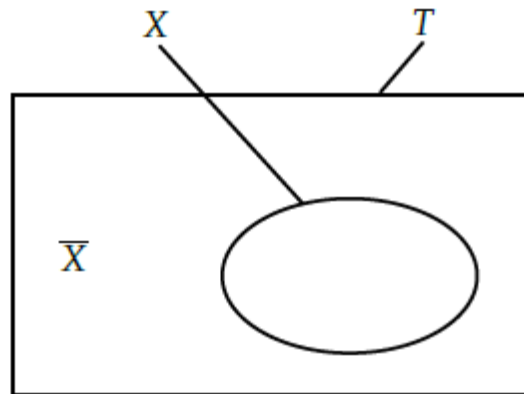


Figure II.1 : Diagramme de Venn de l'ensemble X dans celui des transactions T

II.5.3 Algorithmes :

La technique d'extraction de règles d'association introduite par Agrawal en 1993 est considérée comme l'une des techniques les plus importantes dans la découverte de connaissances dans les bases de données . AIS fut l'un des premiers algorithmes écrits pour les centrales d'achats. Par la suite, en 1994, l'algorithme Apriori est proposé. Deux étapes principales composent ces algorithmes :

1. La recherche des itemsets fréquents ;
2. La génération des règles d'association.

Différentes études soulignent que ce découpage correspond à la manière optimale face à l'explosion combinatoire de l'algorithme. Ainsi, la recherche des itemsets fréquents sur un ensemble m d'attributs binaires est un espace de dimension 2^m . L'étape de génération des règles à partir d'un itemset fréquent de largeur k peut engendrer la génération de $2^k - 2$ règles. [Agr-93]

Depuis l'algorithme de référence d'Agrawal et Srikant, de nombreux algorithmes ont été développés et sont synthétisés dans Hipp et al.. Différents algorithmes ont également été développés pour réduire le nombre d'itemsets en générant des itemsets fermés, maximaux ou optimaux . D'autres algorithmes ont été développés pour réduire le nombre de règles. Enfin, différentes méthodes complémentaires de post-traitement ont été proposées telles que l'élagage (pruning), les résumés (summarizing) ou les regroupements (grouping) [AgS-94].

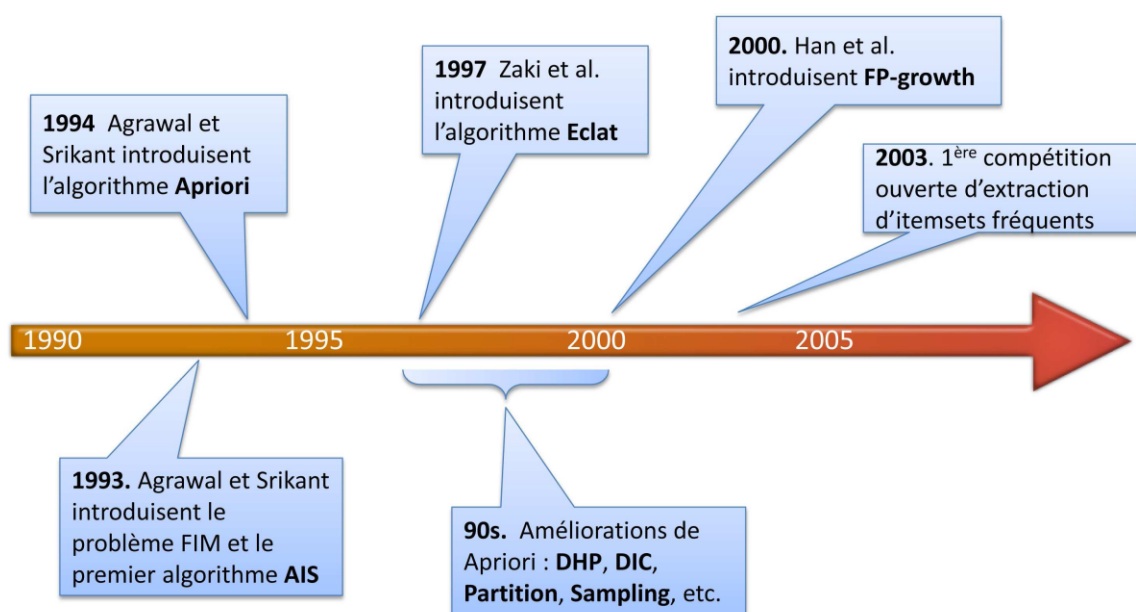


Figure II.2 : Historique des algorithmes d'extraction d'itemsets fréquents

- **Extraction des itemsets fréquents (ou FIM pour Frequent Itemset Mining)**

Les algorithmes d'extraction des itemsets fréquents considèrent un ensemble d'itemsets d'une taille donnée lors de chaque itération, c'est-à-dire un ensemble d'itemsets d'un « niveau » du treillis des itemsets. Ces algorithmes limitent le nombre d'itemsets candidats considérés, en les générant à partir des itemsets fréquents de l'itération précédente : « tous les sur-ensembles d'un itemset infrequent sont infrequent et tous les sous-ensembles d'un itemset frequent sont frequent » .

Les algorithmes **Apriori** et **OCD** réalisent un nombre de balayages du contexte égal à la taille des plus longs itemsets fréquents. L'algorithme **Partition** autorise la parallélisation du processus d'extraction. L'algorithme **DIC** réduit le nombre de balayage du contexte en considérant les itemsets de plusieurs tailles différentes lors de chaque itération. Les algorithmes Partition et DIC entraînent un coût supplémentaire en temps CPU par rapport aux algorithmes Apriori et OCD, dû à l'augmentation du nombre d'itemsets candidats testés. [AgS-94]

- **Extraction des itemsets fréquents maximaux**

Ces algorithmes se basent sur le fait que les itemsets fréquents maximaux, c'est-à-dire les itemsets dont tous les sur-ensembles sont infrequentés, forment une bordure au dessous de laquelle tous les itemsets sont frequent. L'extraction des itemsets fréquents maximaux est réalisée par une exploration itérative du treillis des itemsets fréquents. À partir des itemsets fréquents maximaux, tous les itemsets fréquents sont dérivés et leurs supports sont déterminés en réalisant un balayage du contexte.

Quatre algorithmes basés sur cette approche ont été proposés : **Pincer-Search**, **MaxClique**, **MaxEclat**, et **Max-Miner**. Ces algorithmes permettent de réduire le nombre d'itérations et donc de diminuer le nombre de balayages du contexte et d'opérations CPU réalisées.

- **Extraction des itemsets fréquents fermés**

Les itemsets fréquents fermés sont définis avec la fermeture de la connexion de Galois d'une relation binaire finie. Tous les itemsets fréquents et leurs supports, et donc toutes les règles d'association ainsi que leurs support et confiance, peuvent être déduits efficacement, sans accéder au jeu de données à partir des itemsets fréquents fermés. Les algorithmes **Close** et **A-Close** sont des algorithmes d'extraction des itemsets fermés : ils considèrent un ensemble de générateurs candidats d'une taille donnée, et déterminent leurs support et fermeture en réalisant un balayage du contexte lors de chaque itération. L'algorithme **Close+** permet d'identifier les itemsets fréquents fermés et leurs

générateurs parmi les itemsets fréquents, sans accéder au jeu de données. Notons que tous les itemsets et les itemsets fermés sont fréquents et que tous les itemsets fréquents maximaux sont fermés (cf. figure II.3) [Pas-99].

Han et al. ont récemment classés les algorithmes d'extractions de règles d'association en trois catégories :

1. Les méthodes au format horizontal basées sur Apriori : par exemple Apriori;
2. Les méthodes au format vertical basées sur Apriori : par exemple CHARM;
3. Les méthodes de projections pour le parcours de structures compressées : par exemples FP-growth et FP-tree. [Han-02]

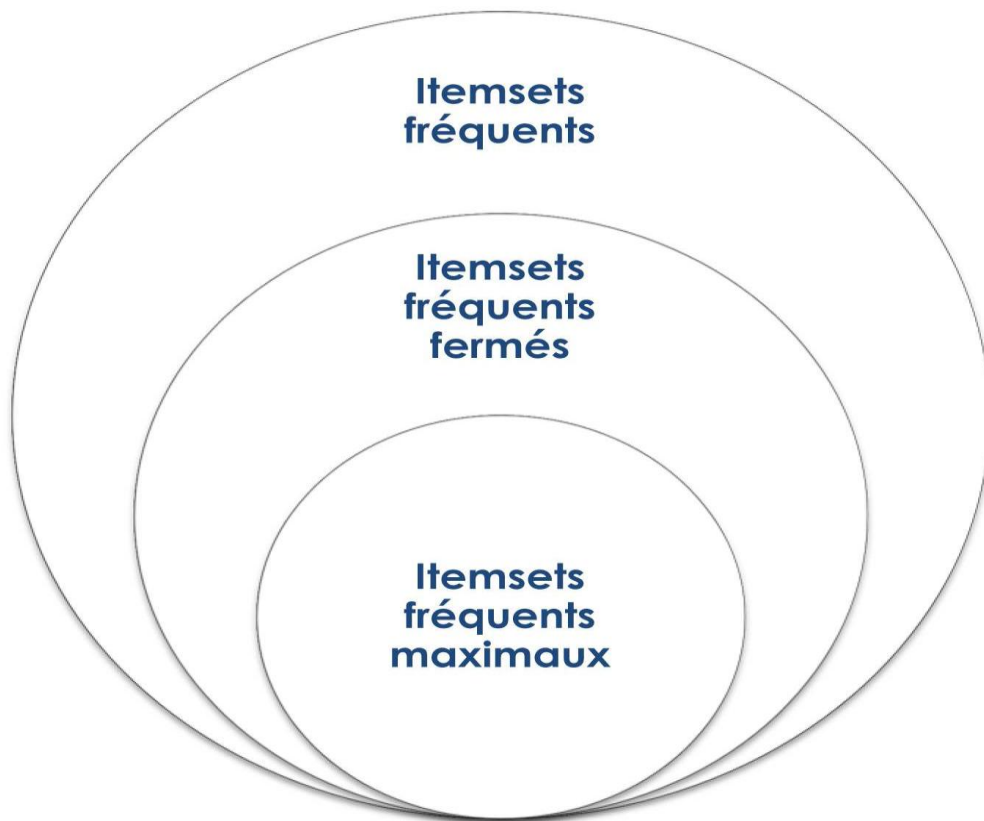


Figure II.3 : Itemsets fréquents, fermés et maximaux

Le paramètre commun de ces trois approches est la détermination du seuil de support minimal assurant la génération des itemsets fréquents. Cette approche présente tout de même un inconvénient : un seuil de support trop faible peut conduire à la génération d'un nombre trop importants d'itemsets tandis qu'un seuil trop important peut minimiser le nombre de règles. Par conséquent, la recherche des top-k itemsets fréquents fermés a été proposée, de longueur minimale $\min_l$, où k est un nombre déterminé par l'expert permettant de fouiller k itemsets fréquents fermés. Top-k fait référence aux k motifs fermés les plus fréquents et $\min_l$ la

longueur minimale des motifs fermés. Pour répondre à cette approche, plusieurs algorithmes ont été proposés, dont TFP (Top-k Frequent Closed Itemsets) dans lequel toutes les transactions plus courtes que min_l sont exclues de la recherche et le support minimal est déterminé dynamiquement lors de la construction du FP-tree, permettant ainsi d'élaguer le treillis avant l'extraction. [Wan-05]

II.5.3.1 La Recherche des Itemsets Fréquents :

La recherche des itemsets fréquents est un problème non trivial car le nombre d'itemsets potentiellement fréquents est exponentiel par rapport au nombre d'items considérés dans la base de données.

On appelle itemset fréquent un itemset dont le support est supérieur à un seuil *minsup* fixé.

Si le support n'est pas atteint, on dit que l'itemset est infrequent.

Soit I , l'ensemble des items, et *minsup* un seuil minimal de support.

L'ensemble F des itemsets fréquents est : $F = \{I \subseteq I \mid \text{support}(I) \geq \text{minsup}\}$

Si $\text{Card}(I) = M$, le nombre d'itemsets possibles est 2^M .

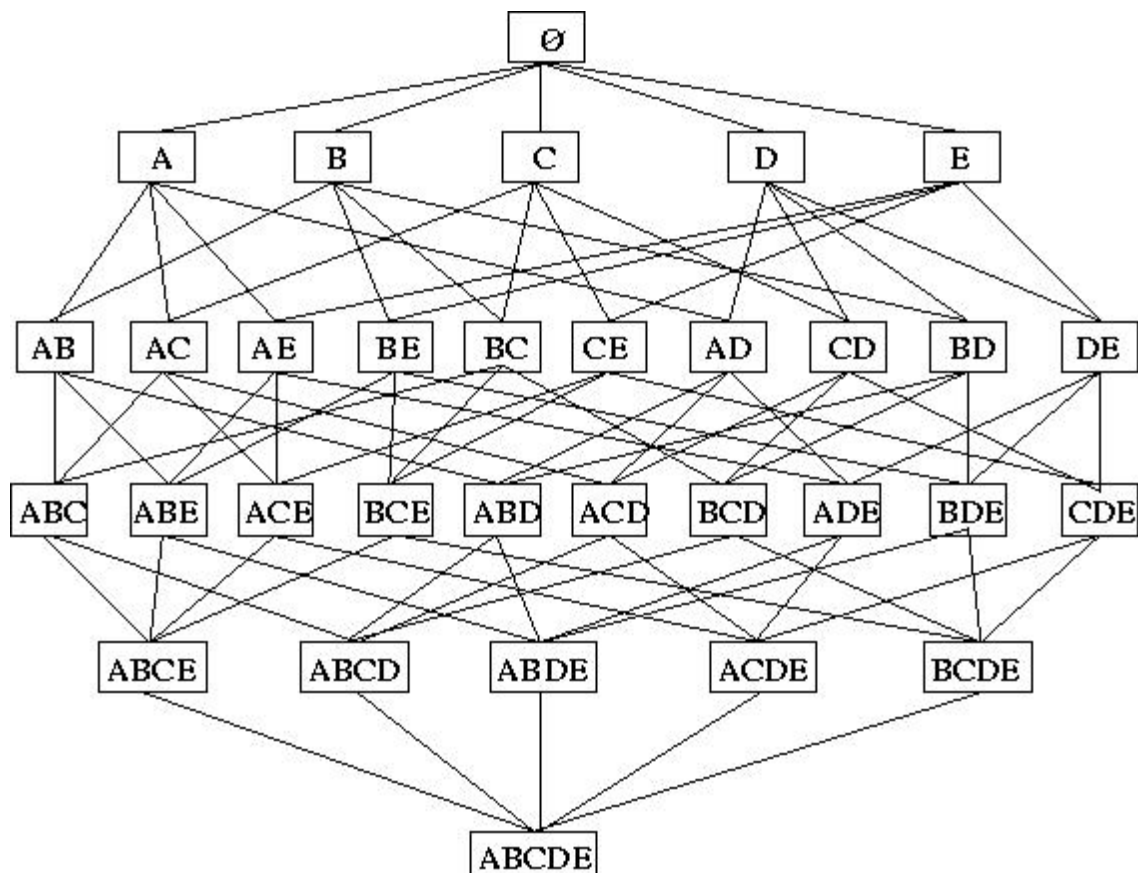


Figure II.4: Treillis d'itemsets.

La figure II.4 présente le treillis des itemsets pour l'ensemble $I=\{A, B, C, D, E\}$. Les itemsets forment un ensemble ordonné, avec l'inclusion comme relation d'ordre. La borne supérieure est définie par l'union et la borne inférieure est définie par l'intersection.

Dans l'exemple, pour seulement 5 items dans I , l'ensemble des itemsets possibles comporte

32 (= 25) éléments (avec l'itemset vide en haut de la figure).

La phase de recherche de ces itemsets fréquents est la phase la plus coûteuse de l'extraction de règles d'association du fait de la taille exponentielle de l'espace de recherche et du nombre élevé nécessaire de balayages complets du jeu de données.

Ces balayages sont nécessaires afin d'évaluer les supports des itemsets (qui permettent de déterminer lesquels sont fréquents) puis de calculer les confiances des règles d'association, mais constituent des opérations très coûteuses en temps d'exécution.

Remarque : Ces deux critères (le nombre de balayages des données et le nombre d'itemsets générés) sont les deux facteurs d'efficacité ou d'inefficacité d'un algorithme de recherche de règles d'association.

L'ensemble des k-itemsets fréquents constituent un treillis partiel (voir Figure II.4). Les supports des itemsets fréquents diminuent de ; vers I , avec $\text{sup}(X) > \text{sup}(XY)$.

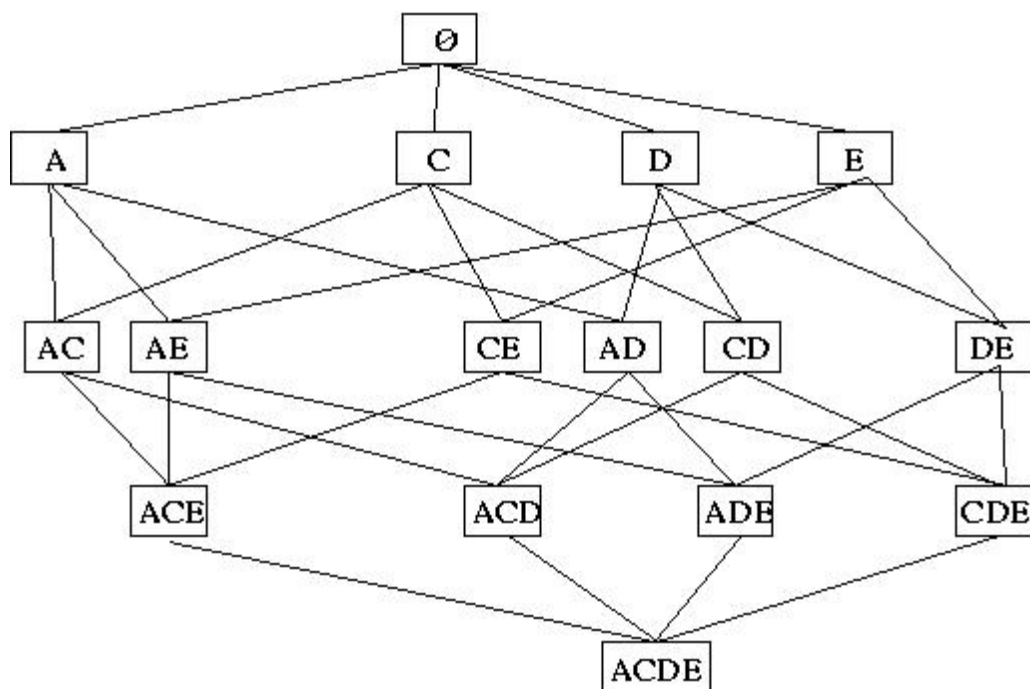


Figure II.4: Treillis partiel composé des itemsets fréquents.

Une approche simpliste consisterait à tester le support de chaque itemset possible. Les cas réels d'application présentent au minimum une centaine d'items et un nombre d'instances en millions (auquel est lié le nombre d'accès disque ou de communications pour calculer les supports) ! De plus, le nombre d'itemsets fréquents est faible par rapport au nombre total d'itemsets.

Il convient donc d'utiliser des approches plus complexes, on peut distinguer deux types d'algorithmes :

- des algorithmes qui proposent une réduction de l'espace de recherche (par exemple l'algorithme de base Apriori).
- des algorithmes qui proposent une réduction du nombre de balayages nécessaires du jeu de données (par exemple l'algorithme DIC).

II.5.3.2 La Génération des Règles d'Association

Pour chacun des itemsets fréquents obtenus, on va rechercher les règles d'association qui possèdent une confiance supérieure au seuil minimal de confiance fourni par l'utilisateur. On regarde les différentes combinaisons condition-conclusion possibles pour chaque itemset. On garde la combinaison la plus pertinente (celle de confiance la plus élevée), si elle répond au critère de seuil minimal de confiance. A partir d'un k-itemset F_k fréquent, on cherche les règles de type : $F_i \Rightarrow F_j$ de support et de confiance suffisants.

F_i et F_j sont des itemsets de tailles respectives i et j , tels que : $i + j = k$, $F_i \subseteq F_k$, $F_j \subseteq F_k$ et $F_i \cap F_j = \emptyset$.

Exemple :

Soit le 3-itemset $\{A B C\}$ déterminé fréquent (les supports des différents itemsets sont indiqués par le Tableau II.5).

Une règle d'association, qui contiendra les trois éléments A, B, C, aura un support de 5%.

itemset	support
$\{A\}$	45%
$\{B\}$	42.5%
$\{C\}$	40%
$\{A B\}$	25%
$\{A C\}$	20%
$\{B C\}$	15%
$\{A B C\}$	5%

Figure II.5:Exemple : les supports des itemsets.

On calcule la confiance des différentes combinaisons possibles. Pour plus de clarté, on ne s'intéresse qu'aux règles dont la conclusion ne comporte qu'un item. Cette limitation est habituelle dans ce type de traitement et permet à la fois d'assurer la lisibilité des résultats obtenus et de limiter l'espace de recherche. Les confiances des combinaisons possibles sont données par le tableau II.2.

La règle apparaissant en dernière ligne est celle de confiance la plus élevée. C'est la plus intéressante des règles générées à partir de l'itemset {A B C}.

Si on fixe le support minimal de confiance à 30%, seule cette règle sera retournée à l'utilisateur (c'est la seule à posséder une confiance suffisante, i.e. supérieure au seuil).

Si par contre, on fixe le seuil de confiance minimal à 40%, aucune règle ne sera renvoyée.

Cette méthode permet de limiter le nombre de règles renvoyées : on ne fournit que les plus pertinentes.

condition	conclusion	confiance
{ A B }	{ C }	20%
{ A C }	{ B }	25%
{ B C }	{ A }	33%

TAB II.2: Exemple : les confiances des différentes règles.

II.6 Mesures de qualité

Les mesures de qualité ou d'intérêts des règles d'association aident les experts à trouver des connaissances intéressantes dans de grands volumes de règles (cf. figure II.7). Ces mesures permettent d'évaluer les règles, de sélectionner celles respectant un seuil de qualité minimale et de les ordonner des plus actionnables au plus irréalisables (cf. figure II.6). Face aux nombreuses mesures d'intérêt présentes dans la littérature, Piatetsky-Shapiro et Matheus proposent de les catégoriser de la manière suivante[Pia-94] :

- Les mesures dites **objectives** (ou data-oriented) dépendantes des données
- Les mesures dites **subjectives** (ou user-oriented) prenant en compte les objectifs et les croyances des utilisateurs.

Cette classification a été revue par Silberchatz et Tuzilin un peu plus tard introduisant une classification des mesures subjectives en mesures actionnables (actionability) et

inattendues (unexpectedness). Différents critères discriminants ont été utilisés pour la définition des mesures d'intérêts :

1. La **Concision** ou la longueur de la règle d'association ;
2. La **Généralité** ou le support de la règle d'association ;
3. La **Fiabilité** ou la confiance de la règle d'association ;
4. La **Validité** ou l'association généralité/fiabilité de la règle d'association ;
5. La **Particularité** ou la manière de gérer les données aberrantes ;
6. La **Diversité** ou la comparaison avec une distribution uniforme ;
7. La **Nouveauté** ou l'apport de connaissances non déductibles par les experts ;
8. L'**Inattendu** ou la contradiction avec les connaissances expertes ;
9. L'**Utilité** ou l'accord avec une fonction d'utilité ;
10. L'**Applicabilité** ou le déclenchement d'une action dans le domaine étudié. [Pia-94]

Geng et Hamilton proposent une catégorisation des mesures d'intérêts sur laquelle nous nous reposerons dans la suite de nos travaux (cf. figure II.6) [Gen-06].

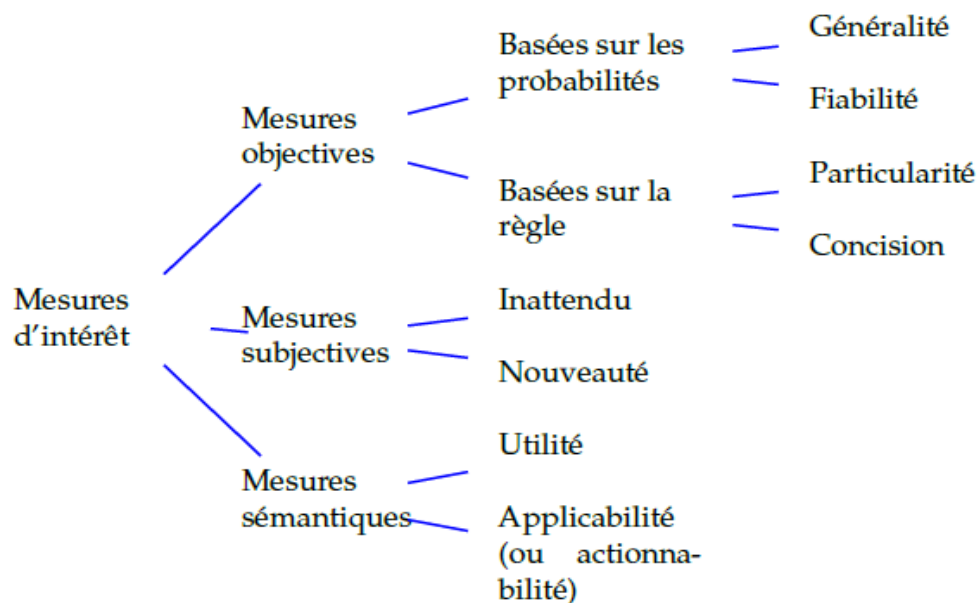


Figure II.6: Catégorisation des mesures d'intérêts des règles d'association [Gen-06].

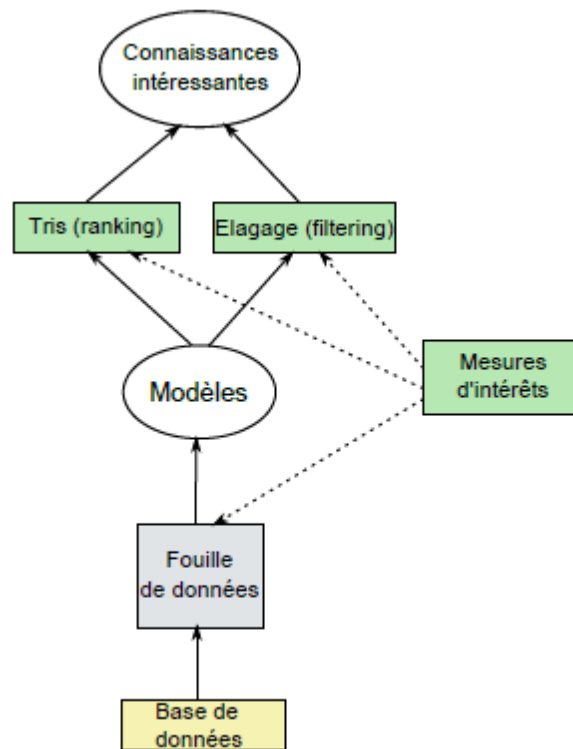


Figure II.7: Rôles des mesures d'intérêts dans le processus d'ECD [Gen-06].

II.6. 1 Mesures objectives

Une méthode populaire pour évaluer l'intérêt des règles d'association consiste à quantifier cet intérêt à l'aide de mesures objectives. Définies à partir de la contingence des règles (caractéristiques des données), les mesures objectives permettent de classer les règles. Cependant, l'utilisation de mesures différentes peut engendrer un classement des règles fortement dissemblable. Ces indices sont adaptés à des particularités de distribution de données ou à des attentes de l'utilisateur et le choix influe directement sur le jeu de règles découvert. La comparaison des indices se base en général sur deux types de critères :

- **Théoriques**, établissant un jeu de caractéristiques attendus et les confrontant avec les propriétés mathématiques des indices.
- **Expérimentaux**, appliquant ces indices sur des jeux de tests et éventuellement utilisant des outils de comparaison de règles d'association.

Les nombreuses comparaisons proposées par Tan et al. montrent qu'aucune mesure ne respecte l'ensemble des critères proposés. Le choix de la mesure dépend fortement des résultats attendus et des paramètres du domaine.

Par exemple, la distribution initiale des données entraîne souvent l'utilisation de mesures différentes. [Tan-04]

Blanchard et al. proposent une classification sémantique basée sur les déviations par rapport à l'indépendance et à l'équilibre et introduisent la qualité des règles d'association au travers de trois concepts : la **généralité** qui peut être calculée par le support et le support causal, la **puissance implicative** qui peut être mesurée par la confiance, l'indice de Loevinger ou la J-Mesure et la **significativité statistique** qui peut être mesurée par l'intensité implication. Les nombreuses mesures existantes et les propriétés de ces mesures ont ainsi suscité un grand nombre de travaux. [Bla-05]

II.6. 1.1 Support et confiance (les limites) :

Définition 1

Le support évalue la généralité de la règle et permet de réduire le nombre de règles à l'aide d'un élagage en fonction d'un seuil minSup. Cependant, l'extraction peut omettre la découverte de « pépites » car généralement, ces dernières, présentent un support faible pour une confiance élevée.

$$\text{support}(X \rightarrow Y) = \frac{n_{xy}}{n}$$

Définition 2

La confiance évalue la validité d'une règle et permet de réduire le nombre de règles à l'aide d'un élagage en fonction d'un seuil minConf. Cependant, cette mesure ne permet pas de détecter l'indépendance des variables. Elle estime la probabilité conditionnelle que la variable Y soit réalisée sachant que la variable X l'est.

$$\text{confiance}(X \rightarrow Y) = \frac{n_{xy}}{n_x}$$

Le support et la confiance sont deux indices élémentaires, mais ils constituent les mesures les plus communément utilisées pour évaluer les règles d'association. Malheureusement, ces indices présentent certaines limites: l'énorme quantité de règles d'association limite l'utilité de la technique et la trivialité d'une grande partie des règles générées implique une réelle nécessité de proposer de nouvelles mesures. [Bri-99]

II.6. 1.2 Le lift (ou l'intérêt)**Définition 3**

La mesure du lift a été introduite par Brin et al. en 1997. Le lift correspond à une mesure d'écart par rapport à l'indépendance. La mesure du lift met en évidence l'intérêt d'une règle d'association comme suit :

$$lift(X \rightarrow Y) = \frac{nn_{xy}}{n_x n_y} = \frac{confidence(X, Y)}{n_y}$$

EXEMPLE Un lift égal à 2 signifie que le nombre d'exemples de la règle $X \rightarrow Y$ est 2 fois plus grand que celui attendu en cas d'indépendance, ce qui veut dire que le consommateur qui achète X a 2 fois plus de chance d'acheter Y qu'un consommateur lambda. Le lift étant symétrique, la situation inverse est vraie puisque les exemples de $X \rightarrow Y$ sont aussi ceux de $Y \rightarrow X$.

La sélection de règles de manière purement statistique (mesures objectives) se heurte à deux écueils principaux : la production d'un trop grand nombre de règles et l'élimination de règles intéressantes si des critères trop restrictifs sont appliqués. [Bri-97]

II.6.2 Mesures subjectives

Les mesures d'intérêts subjectives prennent en compte les objectifs et les connaissances des experts. Il s'agit de méthodes supervisées de traitement des règles suite à leur extraction par des algorithmes classiques, prenant en compte les savoirs et les questionnements des décideurs. Plusieurs approches ont été proposées dans la littérature. Liu et al. définissent les mesures subjectives comme la distance séparant les règles des spécifications des experts. Quant à elle propose davantage une approche en interaction avec les experts en laissant la possibilité de marquer les règles intéressantes. Padmanabhan et Tuzhilin présentent une recherche de règles venant contredire les croyances des experts. Par la suite, les travaux de Silberschatz et Tuzhilin permettent de définir une mesure subjective en comparant les probabilités a priori et a posteriori. Finalement, une approche plus récente de Marinica et Guillet présente une méthode basée sur les schémas de règles et les ontologies.

Liu et al. proposent deux critères subjectifs de sélection de règles : leur caractère inattendu (unexpectedness) et leur actionnabilité (actionability). L'inattendu caractérise le fait que les règles sont intéressantes si elles surprennent les experts métier, c'est-à-dire si elles représentent pour lui une nouveauté. L'actionnabilité mesure l'aptitude d'une règle à être exploitée par l'utilisateur dans le cadre d'une action précise et utile. [Liu-97]

II.6.3 Mesures sémantiques

Selon la classification de Geng et Hamilton, les mesures sémantiques se déclinent en deux catégories :

- Les mesures dites d'utilité : généralement basées sur des notions de « poids » attribués aux items, aux transactions ou aux clients de manière à mettre en valeur l'utilité selon les contraintes du domaine ;
- Les mesures dites d'actionnabilité : généralement orientées métier permettant ainsi d'orienter l'extraction de manière à obtenir des règles pour prendre des décisions.

Le terme « utilité » est utilisé couramment pour définir la « qualité d'être utile » et les utilités sont largement utilisées dans le processus de prise de décision pour exprimer les préférences utilisateurs en fonction des objectifs: décision = probabilité + utilité. [Gen-06]

Selon Shen et al., l'intérêt d'une mesure réside autant dans sa probabilité que dans son utilité. Il présente l'approche Objective-Oriented utilitybased Association (OOA) permettant d'extraire des règles d'association très corrélées aux objectifs et aux utilités des utilisateurs. De plus, il souligne que son approche permettrait aux décideurs de découvrir les meilleurs stratégies ou opportunités en spécifiant les objectifs maximisant les profits et réduisant les coûts. Dans le milieu médical cela pourrait permettre de trouver le meilleur traitement en répondant au besoin en amont « meilleure efficacité et peu d'effets secondaires ». De nombreuses approches mettant en place des mesures d'utilité ont été proposées et sont récapitulées dans le tableau II.3. [She-02]

Mesure sémantique	Modèle	Origine
<i>Weighted support</i>	<i>Weights for items</i>	Support
<i>Normalized weighted support</i>	<i>Weights for items</i>	Support
<i>Vertical weighted support</i>	<i>Weights for transactions</i>	Support
<i>Mixed weighted support</i>	<i>Weights for both items and transactions</i>	Support
OOA	<i>Weights for target attributes</i>	Support
<i>Marketshare</i>	<i>Weight for each transaction</i>	Confiance
<i>Count support</i>	<i>Weights for items and cells in dataset</i>	Support
<i>Amount support</i>	<i>Weights for items and cells in dataset</i>	Support
<i>Count confidence</i>	<i>Weights for items and cells in dataset</i>	Confiance
<i>Amount confidence</i>	<i>Weights for items and cells in dataset</i>	Confiance
<i>Yao et al.'s</i>	<i>Weights for items and cells in dataset</i>	Support

TAB II.3: Exemples de mesures d'utilité

Récemment, Wang et al. suggèrent d'ajouter de la valeur aux règles d'association, par exemple en associant une valeur de marge à chaque item de la base de données, permettant ainsi de trier les règles par gain économique.

Actuellement, les mesures d'utilité connues sont des extensions du support ou de la confiance (cf. tableau II.3). De plus, il existe rarement une mesure d'utilité applicable face à une problématique définie. De nombreuses mesures répondent à un besoin bien précis et sont aujourd'hui non généralisables. [Wan-02]

II.7. Conclusion :

Les règles d'associations tant traditionnelles que séquentielles sont des outils efficaces pour identifier des relations qui existent parmi les données. En plus d'être aisément interprétables, les règles d'associations peuvent faire découvrir aux analystes des associations inattendues.

Ces techniques permettent donc de tirer profit d'une grande quantité de savoir caché dans les données qui pourrait difficilement être découvert autrement. Ces découvertes peuvent ensuite être utilisées et intégrées dans les processus d'affaires de l'entreprise afin d'en améliorer les performances.

Malgré certains problèmes reliés aux temps de calculs et à la taille gigantesque des bases de données à analyser, les algorithmes discutés tout au long de ce texte réussissent tout de même à identifier les règles d'associations présentes dans les données. Comme nous l'avons vu, les règles découvertes peuvent être intra transactionnelles ou intertransactionnelles. La recherche de règles peut également s'effectuer sur une ou plusieurs dimensions, donnant à l'analyste beaucoup de flexibilité pour conduire sa recherche.

La postanalyse est une étape essentielle dans la découverte des règles d'associations puisque sans elle, la compréhension et la synthèse des résultats obtenus seraient des tâches complexes et ardues. Afin de distinguer les règles dignes d'intérêt de celles qui ne le sont pas, diverses mesures objectives et subjectives ont été proposées. Le sujet est toutefois loin d'être clos car le degré d'intérêt d'une règle est fondamentalement subjectif et peut varier d'un domaine à l'autre et même d'un individu à l'autre. Le développement d'outils postanalytiques performant doit se poursuivre, car eux seuls empêchent les analystes d'être noyés sous la multitude de règles d'associations découvertes et rendent la communication des découvertes simple et aisée.

Le chapitre suivant sera consacré à la description détaillée d'un algorithme populaire de règles d'associations qui est l'algorithme A-priori.

Chapitre

III

L'Algorithme A-apriori

III.1. introduction:

Le plus connu des algorithmes de recherche de règles d'association est l'algorithme **Apriori** (Agrawal et al. [Agr-93])

La plupart des algorithmes existants se basant sur Apriori, nous avons décidé de présenter celui-ci en détails.

L'algorithme Apriori est un algorithme permettant de trouver des règles d'association. L'induction des règles d'association, également appelée analyse des paniers d'achats, est une méthode qui a pour objectif de trouver des régularités dans le comportement des consommateurs. Ces régularités se caractérisent par des groupes de produits qui sont régulièrement achetés entre eux. Ainsi, la présence d'un où plusieurs produits dans un panier d'achat peut entraîner la présence d'un ou plusieurs autres produits. Voici un exemple de règles d'association : « Si un consommateur achète du vin et du pain, il achète souvent du fromage également ». Cette règles est relativement prédictible, et ce sont les règles non intuitives qui intéressent principalement les entreprises, c'est-à-dire entre des produits qui semblent n'avoir rien en commun . Or, connaître de telles informations représente un enjeu commercial pour les entreprises : elles peuvent en effet tenter d'augmenter le nombre de produits vendus en disposant stratégiquement les produits en question dans les rayons de magasins (merchandising), ou utiliser ces informations pour la création de prospectus de promotion.

III.2. Principe des algorithmes existants pour la recherche de règles:

Les algorithmes d'extraction des itemsets fréquents procèdent de **façon itérative**, c'est à dire que l'ensemble des itemsets fréquents est parcouru par niveau. Lors de la k^{eme} itération, on recherche les k-itemsets fréquents, c'est-à dire les itemsets de longueur k qui sont fréquents. A l'itération suivante, la $(k + 1)^{eme}$, on recherchera les (k+1)-itemsets fréquents et ainsi de suite.

Les premiers algorithmes d'extraction des itemsets fréquents "par niveaux", **AIS**, **Apriori** et **SETM** ont été proposés en 1993.

Plusieurs autres algorithmes permettant de réduire le temps d'extraction des itemsets fréquents sont ensuite apparus, ainsi que plusieurs optimisations et structures de données permettant d'améliorer l'efficacité de l'algorithme Apriori :

- **AprioriTid** par Agrawal et Srikant – 1994.
- **AprioriHybrid** par Agrawal et Srikant – 1994.
- **DHP** (Direct Hashing and Pruning) par Park et al. - 1995 .
- **Partition** par Savasere et al.- 1995 .
- **Sampling** par Toivonen -1996 .
- **DIC** (Dynamic Itemsets Counting) par Brin et al. - 1997 .

III.3. La notation:

Il existe une notation propre aux règles d'associations et sera examinée brièvement puisqu'elle sera par la suite utilisée dans les prochaines sections. Cette notation, proposée par Agrawal et al. (1993), s'énonce de la manière suivante: soit $I = \{ i_1, i_2, \dots, i_m \}$ l'ensemble des items ou éléments distincts contenus dans une base de données transactionnelles où chaque transaction T est un ensemble d'items de tel sorte que $T \subseteq I$. Une transaction T contient A , un sous-ensemble d'items, si et seulement si $A \subseteq T$. Dans l'exemple de la figure 3.1, $I = \{1, 2, 3, 4, 5\}$ et il y a 10 transactions. [Agr-93]

No_transaction	Items_achetés
10	1, 2, 5
20	2, 4
30	2, 3
40	1, 2, 4
50	1, 2, 3
60	2, 3, 5
70	1, 3
80	1, 2, 3, 5
90	1, 2, 3
100	2, 3

Figure III.1: Exemple d'une base de données transactionnelle contenant 10 transactions.

La notion d'itemset est fondamentale dans les règles d'associations et mérite d'être approfondie. Le terme anglais "itemset" signifie littéralement "ensemble d'items". Un itemset est un sous-ensemble d'items appartenant à l'ensemble I . Une transaction est donc considérée comme un itemset. Les sous-ensembles $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$ et $\{5\}$ de I sont des itemsets de même que $\{1,2\}$ et $\{2,3\}$ ainsi que $\{1,2,5\}$ et $\{1,2,3,4,5\}$. L'itemset $\{1,2\}$, par exemple, a été formé à partir de l'union de l'itemset $\{1\}$ et de l'itemset $\{2\}$. L'itemset $\{1,2\}$ peut ensuite être uni avec l'itemset $\{2,3\}$ pour former l'itemset $\{1,2,3\}$. Les règles d'union dépendent du type de règle d'association recherchée et seront expliquées plus tard. Afin de différencier les itemsets, ceux-ci sont numérotés selon la cardinalité de l'ensemble ou le nombre d'items qu'ils contiennent. Par exemple l'itemset $\{1\}$ est un 1-item set, l'item set $\{1,2,3\}$ est un 3-item set alors qu'un item set composé de k éléments est un k -item set.

Une **règle d'association** est une implication de la forme $A \Rightarrow B$, où A et B sont tous deux des sous-ensembles d'items de I , c'est-à-dire des *itemsets*. La **fréquence** d'un item ou d'un itemset est le nombre de transactions ou d'identifiant unique qui contient au moins une fois l'item ou l'*itemset* en question. Dans l'exemple de la figure 3.1, si $A = \{1\}$ et que $B = \{2\}$ alors la fréquence (A) = 6, la fréquence (B) = 9, et la fréquence de la règle $A \Rightarrow B$ est égale à la fréquence ($\{1,2\}$) = 5. La règle $A \Rightarrow B$ a un **support** s où $s = \text{fréquence}(A \cup B) / n$, n étant le nombre de transactions contenues de la base de données transactionnelles. Ceci peut également s'énoncer sous forme de probabilité $P(A \cup B)$. Dans l'exemple de la figure 3.1, le support de $1 \Rightarrow 2$ est de 50 %, car la fréquence ($\{1, 2\}$) = 5 et $n = 10$. La **confiance**, quant à elle, représente la probabilité conditionnelle

d'obtenir B sachant A ou $P(B \cup A)$. Dans l'exemple, la confiance de $A \Rightarrow B$ est de $5/6$ ou 83,3 %, puisque parmi les 6 transactions où l'item 1 était présent, 5 d'entre-elles contenaient également l'item 2. Toutes les règles d'associations qui dépassent un certain seuil minimal de support (*min_sup*) ainsi qu'un certain seuil minimal de confiance (*min_conf*) sont appelées fortes ou statistiquement intéressantes. Finalement, l'**effet de levier** de la règle $A \Rightarrow B$, ou "lift" en anglais, est donné par $P(B|A)/P(B)$. L'effet de levier indique de combien est multiplié la probabilité d'obtenir B sachant que A fait partie de la transaction. Dans l'exemple de la figure 3.1, l'effet de levier de la règle $A \Rightarrow B$ est de 0,92 (83,3%/90%). Par conséquent, il y a 8 % moins de chance d'obtenir l'item 2 dans une transaction qui contient également l'item 1 que dans l'ensemble des transactions de la base de données. D'autre part, l'effet de levier de la règle $B \Rightarrow C$, où $B = \{2\}$ et $C = \{5\}$ est de $10/9$, ou 1,1, ce qui signifie qu'il y a 10 % plus de chance d'obtenir l'item 5 dans une transaction où l'item 2 est présent que dans l'ensemble des transactions de la base de données. Si l'achat des items 2 et 5 étaient indépendants, $P(5|2) = P(5)$ et par conséquent l'effet de levier serait égale à 1.

III.4. L'algorithme Apriori:

L'algorithme Apriori, introduit par Agrawal et al. (1994) est un algorithme clé pour les règles d'associations car il est à la base de la majorité des algorithmes servant à découvrir des règles d'associations plus complexes telles que les associations séquentielles et multidimensionnelles. Il tire son nom de son heuristique qui utilise l'information connue a priori sur la fréquence des items. Cette heuristique stipule que si A, un sous-ensemble d'items de l'ensemble I, ne possède pas le support minimal, il ne peut être engagé dans une règle d'association avec tout autre item ij de l'ensemble I, $ij \notin A$. Ainsi, si A est peu fréquent la règle $A \Rightarrow ij$ l'est également et il est donc inutile d'examiner toute règle d'association où A est impliqué.

Propriété 1: Tous les sous-ensembles d'un ensemble fréquent sont fréquents.

Cette propriété permet à l'algorithme Apriori de limiter la construction des ensembles "candidats", c'est à dire susceptibles d'être fréquents, de taille k à partir des ensembles fréquents de taille k-1.

Propriété 2: Tous les sur-ensembles d'un ensemble non fréquent sont non fréquents.

Cette propriété permet à l'algorithme Apriori d'éliminer un ensemble candidat si un seul de ses sous-ensembles est non fréquent. [AgS-94]

III.4. 1.Fonctionnement :

Le problème consistant à identifier des règles d'associations se divise en deux étapes :

1. **L'étape de jointure :** Pour trouver les *itemsets* fréquents dans la base de données transactionnelles, l'algorithme *Apriori* effectue plusieurs balayages de la base de données (en anglais *pass*). Le premier balayage sert à identifier les candidats C_k , un ensemble d'*itemsets*, et à compter le nombre de fois qu'apparaît chaque item, c'est-à-dire leur support respectif. Tous les items dont le support est plus grand qu'une valeur prédéterminée appelée *min_sup*, sont conservés afin de former L_k , l'ensemble des k -*itemsets* fréquents. Cet ensemble sert d'amorce pour générer l'ensemble de candidats C_{k+1} .

L'ensemble C_{k+1} , qui regroupe les $(k+1)$ -*itemsets*, est généré en liant L_k avec lui-même. Pour que deux k -*itemsets* puissent être liés, ils doivent posséder $k-1$ items en commun. Par conséquent la liaison de deux 1-*itemsets* requiert aucun élément en commun, alors que le liaison de deux 3-*itemsets* requiert 2 éléments en commun. Les deux 1-*itemsets* {1} et {2} peuvent être liés ensemble pour générer le 2-*itemset* {1,2}. Le 3-*itemset* {1,2,3} peut être lié avec {2,3,4} pour générer {1,2,3,4}, mais ne peut pas être lié avec {3,4,5} puisque seul

l'items {3} est commun aux *itemsets* {1,2,3} et {3,4,5}. Il est fort possible qu'un *itemset* généré ne respecte pas le seuil de support minimal. Si c'est le cas, cet *itemset* est éliminé lors de l'étape d'élagage.

2. **L'étape d'élagage :** Une fois l'ensemble des candidats C_{k+1} généré, le support de tous les $(k+1)$ -*itemsets* est calculé. Tous les $(k+1)$ -*itemsets* $\in C_{k+1}$ dont le support ne dépasse pas le *min_sup* sont retirés de la liste des candidats. Comme la liste des candidats C_{k+1} est réalisée à partir de la liste antérieure des candidats C_k , tout candidat retiré à l'étape k n'est plus considéré dans l'étape $k+1$ comme le stipule l'heuristique d'*Apriori*.

l'algorithme Apriori :

Entrée : la base, le seuil de support minimal

Sortie : l'ensemble F_k des k -itemsets fréquents

```

( 1)  $F_1 = 1$ -itemsets fréquents
( 2) pour( $k = 2$ ;  $F_{k-1} = 0$ ;  $k++$ )faire
( 3)  $C_k = \text{AprioriGen}(F_{k-1})$ 
( 4) pour chaque instance lue inst faire
( 5) pour chaque élément  $c$  de  $C_k$  tel que  $c \in \text{inst}$  faire
( 6)  $c.\text{support}++$ 
( 7) fin pour
( 8) fin pour
( 9)  $F_k = \{c \in C_k | c.\text{support} \geq \text{minsupport}\}$ 
(10) fin pour
(11) retourner  $\cup F_k$ 

```

TAB. III.1 : Principe de l'algorithme Apriori

III.4. 2. Les procédures d'algorithme A-Priori:

III.4. 2.1. La procédure APrioriGen:

La procédure AprioriGen reçoit un ensemble F_{k-1} de $(k - 1)$ -itemsets fréquents comme paramètre. Elle retourne un ensemble C_k de k -itemsets candidats. Les éléments de C_k sont générés par combinaison des éléments de F_{k-1} .

La procédure se décompose en deux étapes, la première visant à générer des candidats, la deuxième visant à en éliminer une partie.

Durant une première partie de la procédure, les $(k - 1)$ -itemsets fréquents de F_{k-1} sont combinés, et les résultats de ces combinaisons sont insérés dans C_k . Durant la deuxième partie, les éléments de C_k dont l'un des sous-ensembles est infrequent sont supprimés de C_k (**élagage** de C_k).

l'algorithme AprioriGen :

Entrée : l'ensemble F_{k-1} des $(k-1)$ -itemsets fréquents

Sortie : l'ensemble C_k des k -itemsets candidats

```

(1) insérer dans  $C_k$ 
(2) select  $p[1], p[2], \dots, p[k-1], q[k-1]$ 
(3) from  $F_{k-1} p, F_{k-1} q$ 
(4) where  $p[1] = q[1], \dots, p[k-2] = q[k-2], p[k-1] < q[k-1]$ 
(5) pour chaque itemset candidat  $c$  de  $C_k$  faire
(6) pour chaque sous-ensemble  $s$  de  $c$  tel que  $|s| = k - 1$  faire
(7) si  $s \in F_{k-1}$  alors supprimer  $c$  de  $C_k$ 
(8) fin pour
(9) fin pour
(10) retourne  $C_k$ 

```

TAB. III.2 – La procédure AprioriGen

III.4. 2.2.Procédure Close, notion d'itemset fermé:

On note que l'itération interne du programme Apriori, qui procède à la construction des ensembles C_k , parcourt l'ensemble B des objets. Elle est coûteuse dans la mesure où elle est interne et dans la mesure où elle conduit à des accès à la base de données contenant B . De nombreux développements ont conduit à une amélioration de l'algorithme Apriori, essentiellement afin de diminuer le coût de ces parcours. Nous présentons ici un de ces développements particulièrement intéressant de ce point de vue, l'algorithme Close. Cet algorithme opère en quelque sorte une abstraction de B en procédant à une extraction d'ensembles fréquents dit "fermés" à partir desquels il est possible de déduire tous les ensembles fréquents. Nous exposons rapidement d'abord la notion de fermeture utilisée, dite fermeture de la connexion de Galois, puis l'extension de la notion de treillis et les propriétés associées aux ensembles fréquents fermés et enfin l'algorithme lui-même.

III.4. 2.2.1. Treillis des ensembles fermés:

Ce treillis est formé d'une façon analogue à celui des itemsets. Il porte sur les itemsets fermés.

Exemple: on trouve dans la Figure III.2 ce treillis (en caractère gras) superposé au treillis des itemsets.

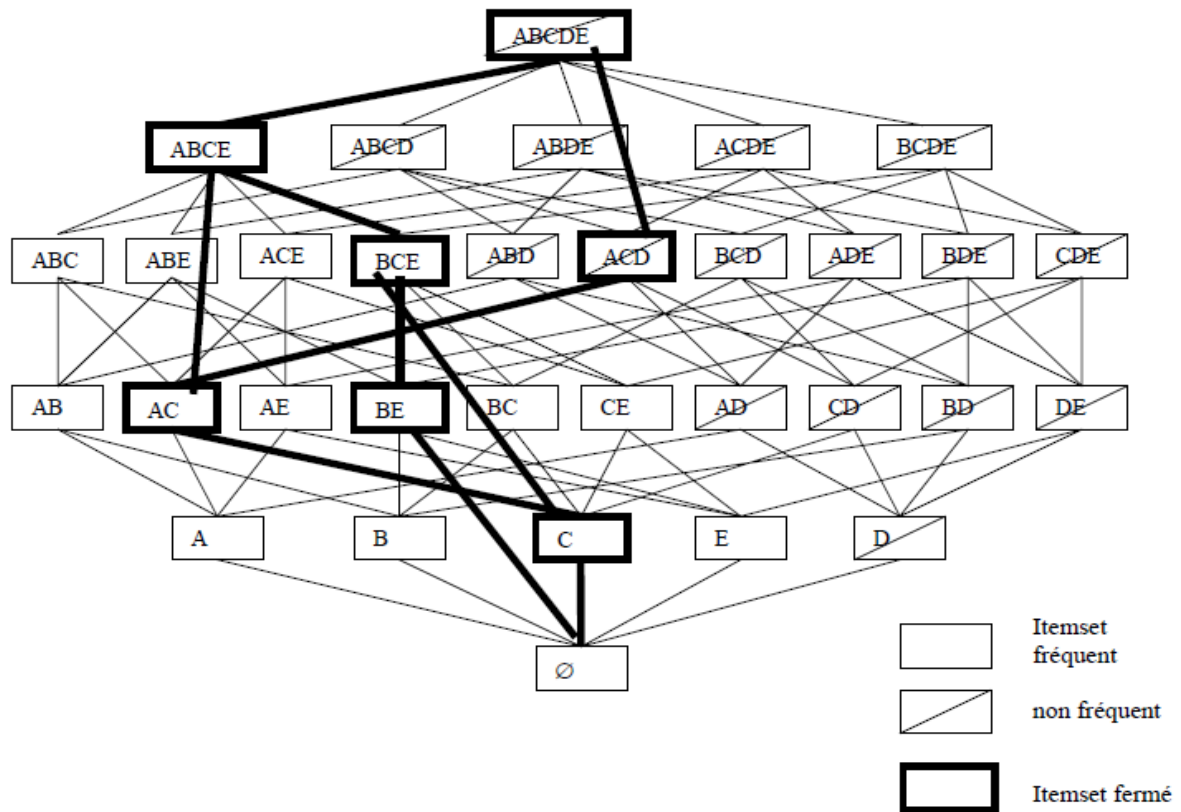


Figure III.2 : Treillis des ensembles fermés relatif à B.

De même, la notion d'ensemble fermé fréquent est une extension de celle d'ensemble fréquent

Exemple:

BCE est fréquent car $\text{fréquence}(\text{BCE}) = |\psi(\{B, C, E\})| / |B| = 4/6 > \text{minfréquence}$. On définit aussi par extension l'ensemble des ensembles fermés fréquents FF et aussi l'ensemble des ensembles fermés fréquents "maximaux" FM:

$$\text{FM} = \{ i \subseteq I \mid i \in \text{FF} \wedge (\forall i' \supset i, \text{fréquence}(i') < \text{minfréquence}) \}$$

III.4. 2.2.2. L'Algorithme Close:

L'approche est fondée sur l'idée suivante:

-1ère étape: détermination de FF

2ème étape: déduire les ensembles fréquents à partir de FF par production de tous les sous-ensembles des éléments de FF et déterminer leur support à partir de leur fermeture.

C'est la première étape qui est cruciale. Ses performances doivent être meilleures pour l'extraction des ensembles fermés fréquents qu'Apriori pour l'extraction des ensemble fréquents. L'idée essentielle est d'utiliser un critère supplémentaire par rapport à Apriori pour réduire encore plus les itemsets fréquents candidats (dans ce cas, ils sont aussi fermés).

La mise en oeuvre de Close utilise la notion d'itemset "générateur" pour produire les candidats. Un itemset générateur d'un itemset fermé f est un itemset minimal (au sens de l'inclusion) dont la fermeture est f

début

%Un k-générateur est une structure à trois champs: générateur (un k-itemset), fermeture (celle du générateur), fréquence (celle du générateur, identique à celle de la fermeture).

FFk est l'ensemble des k-générateurs fréquents.

FFCk est l'ensemble des k-générateurs candidats issu de FFk-1 .

La notation FFCk .générateurs désigne l'ensemble des itemsets générateur des kgénérateurs

de FFC.

L'union des ensembles FFk est le résultat attendu. %

```
FFC1. générateurs := I ;
k := 1 ;
tantque FFCk. générateurs ≠ ∅
faire FFCk := Gen-Closure(FFCk, B) ;
% les champs fermeture et support des éléments de FFCk sont mis à jour %
FFk := ∅ ;
pour chaque c ∈ FFCk
faire si c.fréquence ≥ minfréquence alors FFk := FFk ∪ {c} fsi fait;
FFCk+1. générateurs := Gen-Generator(FFk) ;
% les champs générateur des éléments de FFCk+1 sont mis à jour à partir de FFk en
utilisant les propriétés 1et 2 et le critère cité en 2.2.d.3 %
k := k + 1
fait
fin
```

Figure III.3 : Extraction des ensembles fermés fréquents à l'aide du programme Close

III.4. 2.3.Procédure de génération des règles d'association à partir des EIF:

Disposant des EIF(Ensemble de Items Fréquents) , il nous faut maintenant les transformer en règles.

$e \in \{\text{ensemble de items fréquents}\}$, $\text{card}(e) \geq 2$

- générer les sous-ensembles h de e , $h \neq \emptyset$, $h \neq e$
- calculer $c = \text{support}(e)/\text{support}(e - h)$;
- si $c \geq \text{minconf}$ alors générer la règle $(e - h) \rightarrow h$.

ensemble items fréquents

$e = \{ABC\}$

sous-ensemble h	règle $(e - h) \rightarrow h$
$\{A\}$	$BC \rightarrow A$
$\{B\}$	$AC \rightarrow B$
$\{C\}$	$AB \rightarrow C$
$\{AB\}$	$C \rightarrow AB$
$\{AC\}$	$B \rightarrow AC$
$\{BC\}$	$A \rightarrow BC$

TAB. III.3: ensemble items fréquents $e = \{ABC\}$

III.4. 2.3.1 Réduction du nombre de règles

- $\text{support}(A) > \text{support}(A;B)$
- $\text{support}(A;B;C) = \text{support}(A) \wedge \text{support}(A;B;C) = \text{support}(A;B)$
- $\text{confiance}(A \rightarrow BC) \wedge \text{confiance}(AB \rightarrow C)$

$$A \rightarrow BC \text{ valide} \Rightarrow \left. \begin{array}{l} AB \rightarrow C \\ AC \rightarrow B \end{array} \right\} \text{valides}$$

$$AB \rightarrow C \text{ non valide} \Rightarrow \left. \begin{array}{l} A \rightarrow BC \\ B \rightarrow AC \end{array} \right\} \text{non valides}$$

- Générer l'ensemble H des singletons de e
- Pour chaque $h \in H$, calculer $c = \text{support}(e) - \text{support}(e - h)$
- Si $c > \text{minconf}$ alors ajouter la règle $r : (e - h) \rightarrow h$
- Sinon enlever h de l'ensemble H
- Générer un nouvel ensemble H d'ensembles de cardinal 2

III.4. 2.3.2. Ensemble de items fréquents:

$\{ABC\}$

$H = \{\{A\}, \{B\}, \{C\}$

sous-ensemble	règle
$\{A\}$	$BC \rightarrow A$
$\{B\}$	$AC \rightarrow B$
$\{C\}$	$AB \rightarrow C$

Si $BC \rightarrow A$ non valide, $H = \{\{B, C\}\}$

sous-ensemble	règle
$\{BC\}$	$A \rightarrow BC$

Procédure Gen- règles(E ; minsup; minconf)

Entrée : E : ensemble d'ensembles de items , $minsup$, $minconf$: entier

```

1 Début
2   pour chaque  $e \in E$ ,  $card(e) \geq 2$  faire
3      $m \leftarrow 1$ 
4      $H \leftarrow \{\text{singletons sous-ensemble de } e\}$ 
5     tant que  $m \leq card(e)$  faire
6       pour chaque  $h \in H$  faire
7          $confiance(r) \leftarrow support(e)/support(e - h)$ 
8         si  $confiance(r) \geq minconf$  alors
9            $R \leftarrow R \cup \{(e - h) \rightarrow h\}$ 
10        sinon  $H \leftarrow H \setminus \{h\}$ 
11       $H \leftarrow \text{APRIORI-GEN}(H)$ 
12       $m \leftarrow m + 1$ 
13  Retourner  $R$ 
14 Fin

```

TAB. III.4 : Procédure Gen- règles

III.5.Exemple détaillé de l'algorithme Apriori:

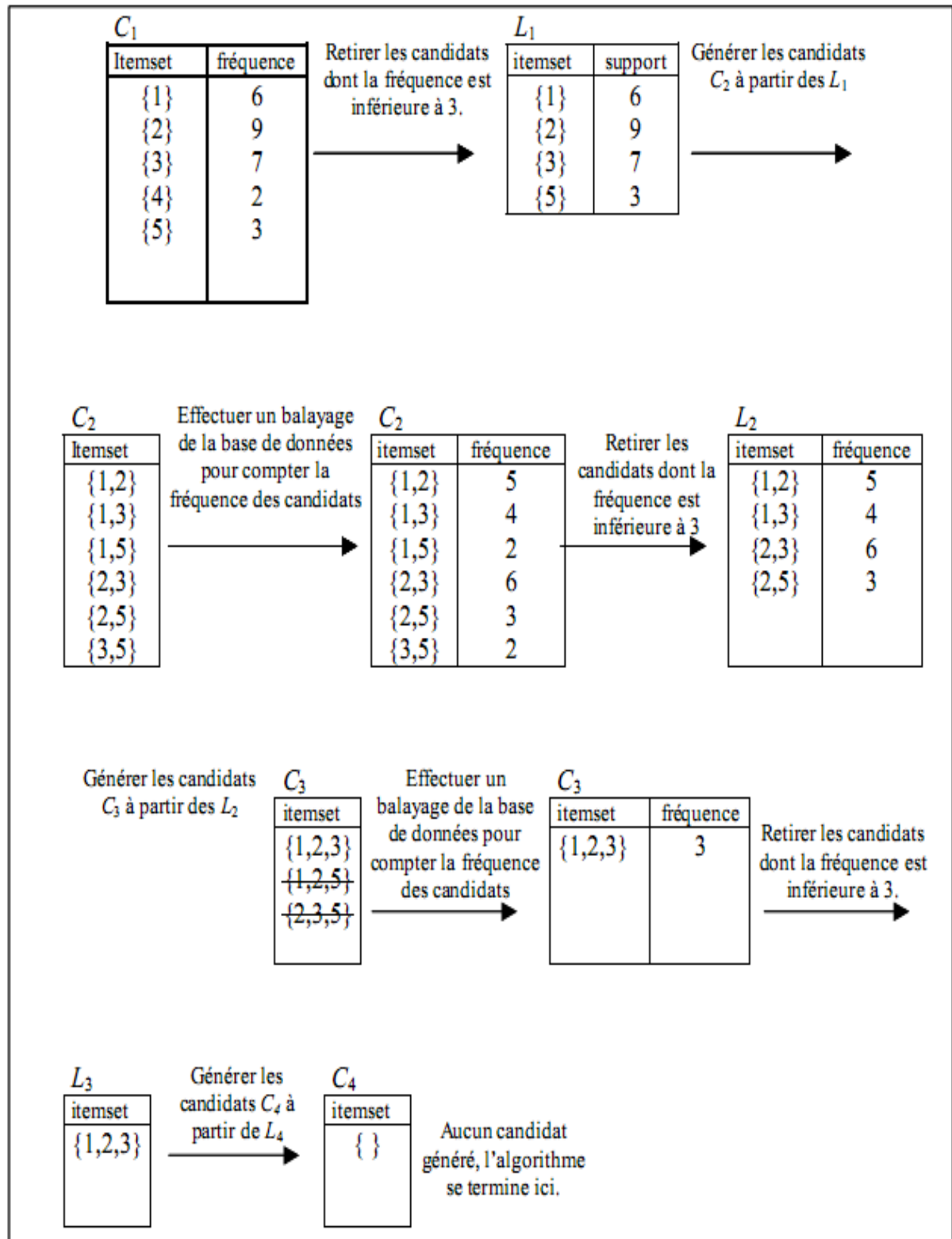
Voici un exemple détaillé des étapes suivies par l'algorithme *Apriori*. La base de données utilisée pour cet exemple est celle qui se retrouve à la figure 3.1. Cette base de données contient 10 transactions et pour cet exemple, le support minimal est fixé à 30 % soit un décompte minimal requis de 3 transactions. Les résultats détaillés de chacune des étapes sont illustrés à la figure 3.2

1. Lors de la première itération, l'algorithme compte le support de chaque 1-*itemset* de la base de données. Ces *itemsets* forment l'ensemble des candidats C_1 qui sert à générer l'ensemble L_1 , c'est à dire l'ensemble des 1-*itemsets* fréquents.

2. Premier élagage. L'algorithme compare ensuite la fréquence de chaque 1-*itemset* avec le support minimal prédéterminée ici à 30 % ou une fréquence de 3. Tous les *itemsets* ayant une fréquence inférieure à 3 sont retirés et ceux dont la fréquence est supérieure ou égal à 3 sont conservés afin de générer l'ensemble L_1 . Les 1-*itemsets* fréquents qui forment L_1 sont dans ce cas-ci {1}, {2}, {3} et {5}.

3. Étape de jointure. Les 1-*itemsets* de l'ensemble L_1 sont utilisés pour générer les candidats C_2 . La génération des candidats est réalisée en liant l'ensemble L_1 avec lui-même. Comme il s'agit de 1-*itemsets*, le nombre de combinaison possible est de $n(n-1)/2$, n étant le nombre d'*itemsets*. Dans cet exemple, le nombre d'*itemsets* étant de 4, six candidats sont formés. Les candidats sont {1,2}, {1,3}, {1,5}, {2,3}, {2,5} et {3,5}.

4. Calcul du support. Lorsque les 2-*itemsets* candidats ont été générés, l'algorithme effectue un autre balayage de la base de données afin de calculer la fréquence respective des candidats. Cette fréquence est inscrite dans une table comme illustré à la figure 3.2.

Figure III.3 : Étapes de l'algorithme *Apriori* avec un support minimal de 30

5. Deuxième élagage. L'algorithme va par la suite parcourir l'ensemble C_2 afin d'éliminer tous les 2-*itemsets* ayant une fréquence inférieure à 3. Dans ce cas-ci, les 2-*itemsets* $\{1,5\}$ et $\{3,5\}$ sont retirés. Les autres 2-*itemsets* sont conservés et forment l'ensemble L_2 , c'est-à-dire l'ensemble des 2-*itemsets* fréquents.

6. Génération des candidats. La génération des 3-*itemsets* est réalisée en fusionnant L_2 avec lui-même. Comme cette fusion implique des 2-*itemsets*, les *itemsets* doivent avoir 1 *itemset* (2-1) en commun. De plus, comme le stipule l'heuristique *Apriori* tous les sous-ensembles de (k-1)-*itemsets* formés doivent être fréquents. Si un *k-itemset* généré est composé de (k-1)-*itemsets* non fréquents, celui-ci est automatiquement éliminé, ce qui évite de calculer son support ou sa fréquence. Dans l'exemple de la figure 2.2, $\{1,2\}$ et $\{2,5\}$ peuvent être fusionnés pour former l'*itemset* $\{1,2,5\}$. Les sous-ensembles de $\{1,2,5\}$ sont $\{1,2\}$, $\{1,5\}$ et $\{2,5\}$. Comme $\{1,5\}$ ne figure pas parmi l'ensemble L_2 des 2-*itemsets* fréquents, l'*itemset* $\{1,2,5\}$ ne peut pas avoir une fréquence supérieure à 3. Il est donc retiré des candidats et son support n'est pas comptabilisé. L'*itemset* $\{2,3,5\}$ est également élagué car $\{3,5\}$ ne figure pas parmi l'ensemble L_2 2-*itemsets* fréquents. À la fin de cette étape, seul l'*itemset* $\{1,2,3\}$ est généré.

7. Calcul du support. Un troisième balayage de la base de données sert à calculer la fréquence de l'*itemset* $\{1,2,3\}$, et cette dernière est de 3.

8. Étape d'élagage. L'algorithme compare la fréquence de l'*itemset* $\{1,2,3\}$ avec la fréquence minimale. Comme $\{1,2,3\}$ possède la fréquence minimale, celui-ci est conservé et devient le seul *itemset* de L_3 , l'ensemble des 3-*itemsets* fréquents.

9. Génération des candidats. Étant donné que l'ensemble d'amorce L_3 ne contient qu'un seul *itemset*, soit $\{1,2,3\}$, aucun 4-*itemset* candidat ne peut être généré. L'algorithme se termine ici.

Les associations identifiées par l'algorithme sont celles formées par les *itemsets* de l'ensemble L_2 et de l'ensemble L_3 soit $\{1,2\}$, $\{1,3\}$, $\{2,3\}$, $\{3,5\}$ et $\{1,2,3\}$. Ces *itemsets* fréquents engendrent les règles d'associations. Ainsi, l'*itemset* $\{1,2\}$ engendre les règles d'associations « $1 \Rightarrow 2$ » et « $2 \Rightarrow 1$ » alors que l'*itemset* $\{1,2,3\}$ engendre les associations

« $1 \Rightarrow 2 \Rightarrow 3$ », « $1 \Rightarrow 3 \Rightarrow 2$ », « $2 \Rightarrow 1 \Rightarrow 3$ », « $2 \Rightarrow 3 \Rightarrow 1$ », « $3 \Rightarrow 1 \Rightarrow 2$ »
et « $3 \Rightarrow 2 \Rightarrow 1$ ».

III.5.Complexité de base de la génération d'itemsets fréquents:

Le principe des algorithmes de base de génération des itemsets fréquents algorithmes est itératif et consiste à répéter les actions :

- utiliser les $(k - 1)$ -itemsets pour générer les k -itemsets candidats ;
- scanner les instances pour calculer les supports de candidats.

La complexité de la génération de ces itemsets peut se décomposer en deux composantes:

- le nombre d'itemsets à générer ;
- le calcul de support pour les k -itemsets.

Nombre d'itemsets générés

Le nombre d'itemsets possibles est 2^m .

Le principe de l'algorithme itératif permet de ne pas générer la totalité des itemsets, on n'examine donc pas 2^m itemsets, mais beaucoup moins.

Pour chaque niveau k , on ne génère donc pas tous les k -itemsets mais seulement un sous- ensemble, basé sur le nombre de $(k - 1)$ -itemsets fréquents identifiés au niveau précédent (voir l'exemple présenté plus loin).

Pour m items, on identifie m 1-itemsets candidats. Le calcul des supports de ces candidats via un examen de l'ensemble des données permet d'identifier m' 1-itemsets fréquents (avec $m' < m$).

Le nombre de 2-itemsets possibles est $C_m^2 = m!/(m - 2)!2! = m(m - 2)/2$

Or, on ne génère qu'un sous-ensemble des 2-itemsets en ce basant sur la propriété : (Tous les sous-ensembles d'un itemset fréquent sont fréquents).

Ainsi, on génère les candidats de taille 2 à partir des m' 1-itemsets fréquents, soit $m'(m' - 2)/2$, 2-itemsets candidats: $m'(m' - 2)/2 < m(m - 2)/2$ puisque $m' < m$. De plus, on ne poursuit pas obligatoirement la génération jusqu'au itemsets de taille m , on s'arrête à la génération des l -itemsets (où $l < m$). Le critère d'arrêt de l'algorithme est lié à l'impossibilité de générer des candidats de taille $l + 1$ (dans l'exemple présenté plus loin, on s'arrête à $l = 4$).

Nombre de scans des données

Le nombre de scans nécessaires des données dépend de l'implémentation réalisée. Pour calculer le support des 2^k -itemsets possibles, on doit réaliser :

- 2^k scans de la base, un pour chacun des itemsets, dans la version de base de l'algorithme (on effectue un scan des données pour chaque itemset dont on veut calculer le support) ;
- $k - 1$ scans de la base, un pour chaque itération sur k dans une version utilisant une structure de treillis pour stocker les itemsets (on effectue un scan des instances à chaque itération et on fait parcourir la structure à chacune des instances) ;
- moins de $k - 1$ scans de la base dans des algorithmes visant à optimiser encore cette étape .

En considérant, la version de base, nécessitant un scan des données pour chaque itemset, la complexité de la phase de génération des itemsets fréquents est donc de l'ordre de $O(n2^m)$ (où n est le nombre d'instances, m le nombre d'items).

III.6.Améliorations proposées à Apriori :

- **DHP** (Direct Hashing and Pruning) proposé par Park et al.

DHP est une légère modification de Apriori qui utilise des tables de hachage pour diminuer le nombre d'itemsets candidats générés.

Cet algorithme propose également une diminution de la taille de la base de données, au fur et à mesure des itérations. Lors des deux premières itérations (gestion des 1-itemsets et des 2-itemsets), on utilise une table de hachage sur les numéros des itemsets . A chacune des cases de la table de hachage, on associe un compteur. A chaque fois qu'une instance contient l'un des itemsets associés (par la fonction de hachage) à une case de la table de hachage, on incrémente le compteur de cette case. On garde les itemsets associés aux cases dont le compteur est suffisant (pour une case associée à 3 itemsets, on garde les itemsets si le compteur associé à la case possède une valeur supérieure à trois fois le seuil de support fixé). A partir de la deuxième itération, on effectue un pruning (voir page 104) des instances de la base. Pour chaque instance, si le nombre de bits de cette instance est supérieur à k , l'instance peut être utile à la k^{eme} itération, sinon, on la supprime.

- **AprioriTid** proposé par Agrawal et Srikant .

Cet algorithme utilise des listes d'itemsets associées aux identifiants des objets en relation avec les itemsets, afin de diminuer progressivement la taille du jeu de données. Il permet de diminuer progressivement la taille de la base, dans le but de la stocker en mémoire et de ne plus réaliser d'opérations d'entrée-sortie, après le premier balayage.

A partir de la deuxième itération, la procédure AprioriGen est utilisée pour générer les itemsets candidats de l'itération suivante dans un ensemble C_k . Chaque élément de l'ensemble C_k est un couple (ID, c_k) dans lequel c_k est la liste des k-itemsets candidats contenus dans l'instance d'identifiant ID. Le support d'un itemset candidat c_k correspond au nombre d'apparitions de c_k dans C_k . [AgS-94]

- **AprioriHybrid** proposé par Agrawal et Srikant .

AprioriHybrid constitue un mélange de Apriori et AprioriTid en fonction de l'itération à laquelle on se trouve. [AgS-94]

- **Partition** proposé par Savasere et al.

Cet algorithme ne nécessite que deux balayages du jeu de données.

Durant le premier balayage, la base est divisée en n partitions disjointes qui seront considérées une à une successivement. Pour chaque partition, l'ensemble des itemsets fréquents dans la partition (les itemsets fréquents locaux) sont extraits. Les ensembles d'itemsets fréquents de chaque partition sont ensuite fusionnés pour obtenir un ensemble d'itemsets candidats à être fréquents sur la base entière. On a ainsi un sur-ensemble de l'ensemble des itemsets fréquents.

Durant le second balayage les supports pour ces candidats sont calculés sur toute la base, les itemsets fréquents sont identifiés. La taille de la partition est choisie de telle façon que chaque partition tienne en mémoire[Sav-95].

- **MaxEclat / MaxClique** proposé par Zaki et al.

A chaque itemset on associe une tid-liste (liste des identifiants des instances qui contiennent cet itemset). Ces algorithmes effectuent ensuite des jointures sur les tid-listes des instances associés à chaque itemset. Le nombre d'instances contenus dans une tid-liste est le support de l'itemset associé à cette tid-liste. [Zak-02]

- **Sampling** proposé par Toivonen

Cet algorithme effectue la génération des itemsets sur un échantillon de la base. Leurs supports sont ensuite calculés sur la base entière. Au fur et à mesure du parcours de la base, on affine les résultats (la valeur des supports). On supprime des itemsets qui se révèlent non fréquents.

Cet algorithme suppose de disposer d'un échantillon représentatif de la base !

- **DIC** (Dynamic Itemsets Counting) proposé par Brin et al.

Cet algorithme propose une diminution du nombre de balayages nécessaires du jeu de données. L'algorithme DIC est basé sur une méthode similaire à Apriori. Le jeu de données est décomposé en sous-ensembles de taille donnée, et durant chaque itération, un sous-contexte (i. e. un sous-ensemble d'instances) est lu. Ainsi, plusieurs ensembles d'itemsets candidats de tailles différentes sont traités simultanément durant chaque itération, ce qui permet de diminuer le nombre total de balayages du contexte.

L'algorithme utilise un principe de fenêtrage de la base (on forme des blocs de M instances). On affecte un type aux itemsets : fréquent confirmé, fréquent potentiel, infrequent confirmé, infrequent potentiel. Le parcours du premier bloc permet d'estimer les fréquences des 1-itemsets. Lorsqu'on a parcouru tout ce premier bloc, on génère les 2-itemsets candidats à partir des estimations de fréquences réalisées pour les 1-itemsets. Le parcours du deuxième bloc permet d'améliorer les estimations des fréquences des 1-itemsets et de débiter l'estimation des fréquences des 2-itemsets... [Bri-97]

III.7.conclusion :

L'algorithme A-priori est notamment appliquée au problème de l'analyse du panier de la ménagère. Dans ce cas, chaque individu est un panier d'un client de supermarché et chaque item est un produit qui s'y trouve. Le gérant du supermarché est intéressé par savoir quels articles sont achetés ensemble de manière fréquente.

Pour limiter le nombre d'attributs (un attribut = un type de produit vendu), on peut effectuer des regroupements d'items comme << conserves>> , << conserves de légumes>> << conserves de légumes de telle marque >>, << conserves de légumes de telle taille >>

On peut aussi ajouter d'autres attributs virtuels comme <<fin de semaine >>ou <<période de vacances scolaires >> qui permettent d'extraire des règles de la forme : << si produits a, b et c et fin de semaine alors produit l >>.

Dans d'autres applications d'extraction de règles d'association, les individus sont des textes et les items des mots, ou encore les individus sont être des phrases et les items des documents.

Dans ce travail nous avons l'implémentation l'algorithme A-priori et applique à base de donnée d'un pharmacie pour extraire des connaissances permettes de faire une meilleur gestion de ses opérations .alors que le pharmacien est intéressé par savoir quels médicaments sont achetés ensemble de manière fréquente.

Chapitre

IV

Conception, mise en œuvre

IV.1. Introduction :

Afin de mettre en œuvre un logiciel, il faut d'abord étudier son cycle de vie (en anglais software lifecycle), qui désigne toutes les étapes de développement d'un logiciel, de sa conception à sa disparition. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la vérification du processus de développement, c'est-à-dire l'adéquation des méthodes mises en œuvre.

Notre conception sera confirmée aux activités suivantes faisant référence à un modèle connu dit modèle en cascade.

- Analyse des besoins : c'est-à-dire l'expression, le recueil et la formalisation des besoins et des objectifs attendus.
- Spécification fonctionnelle : il s'agit de définir l'architecture logicielle de système en présentant les principales fonctionnalités à implémenter.
- Découpage modulaire (conception globale) : il s'agit de l'élaboration de tous les modules nécessaires.
- Conception détaillée : consistant à définir précisément le fonctionnement algorithmique de chaque module.
- Codage (implémentation ou programmation) : soit la traduction dans un langage de programmation des fonctionnalités définies lors de la phase de conception.

Tests : permettant de vérifier individuellement que chaque sous-ensemble du logiciel est implémenté conformément aux spécifications.

IV.2. conception :

IV.2.1. Analyse des besoins:

Nous avons comme objectif principal la réalisation d'un système d'aid à la décision qui réunit quelques exigences de fiabilité.

Nous avons spécifié ses exigences par les points suivants :

- Avoir un système d'aide à la décision fiable .
- Avoir un système capable pour une représentation efficace des données.
- Valider notre système par des tests en discutant ses performances .

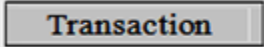
IV.2.2. Spécification fonctionnelle

La conception sera modélisée à l'aide de langage UML(Unified Modeling Language) qui est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique. [2]

IV.2.2.1. Les acteurs du système

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. [Pas07]

Le tableau suivant présente les différents acteurs de notre système.

Acteur	Description
 Utilisateur	L'administrateur a pour mission : - Lancer le extraction - Etablir des statistiques
 BDD	Contient un ensemble des transactions (ordonnances)
 Transaction	Objet pour l'extraction

TAB IV.1. Les acteurs du système.

IV.2.2.2. Détermination des cas d'utilisation

Un *cas d'utilisation* (use case) représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier.
[Pas-07]

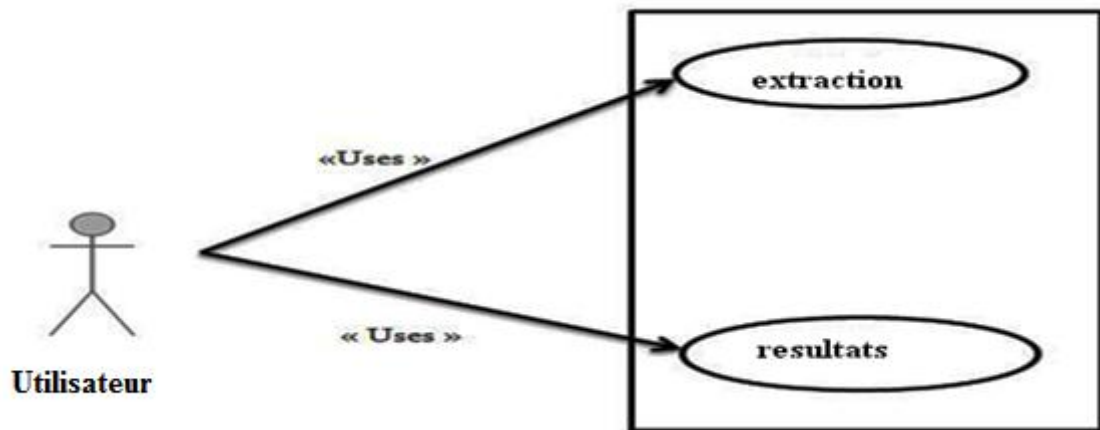


Figure IV.1. Diagramme de cas d'utilisation de système.

Description des cas d'utilisation

Description de cas d'utilisation de « Extraction »

Titre : extraction des connaissances à partir de base de données .

Description : ce cas d'utilisation permet à l'administrateur de faire un extraction des Regles d'associtons . Lors de cette phase le système extrait les ensembles fréquents et rend le résultat de l'extraction.

Acteurs : administrateur, transaction(ordonnance).

Description de cas d'utilisation « resultats »

Titre : Resultats

Description : ce cas d'utilisation permet à l'utilisateur d'établir des resultats sur le système, en effet cette phase nous permet d'afficher differente résultats de l'extraction de programme.

Acteur : l'administrateur, base de transactions.

IV.2.2.3. Diagramme de séquence de cas d'utilisation « Extraction »

Dans ce cas, on présente la séquence des scénarios du cas d'utilisation «Extraction ».

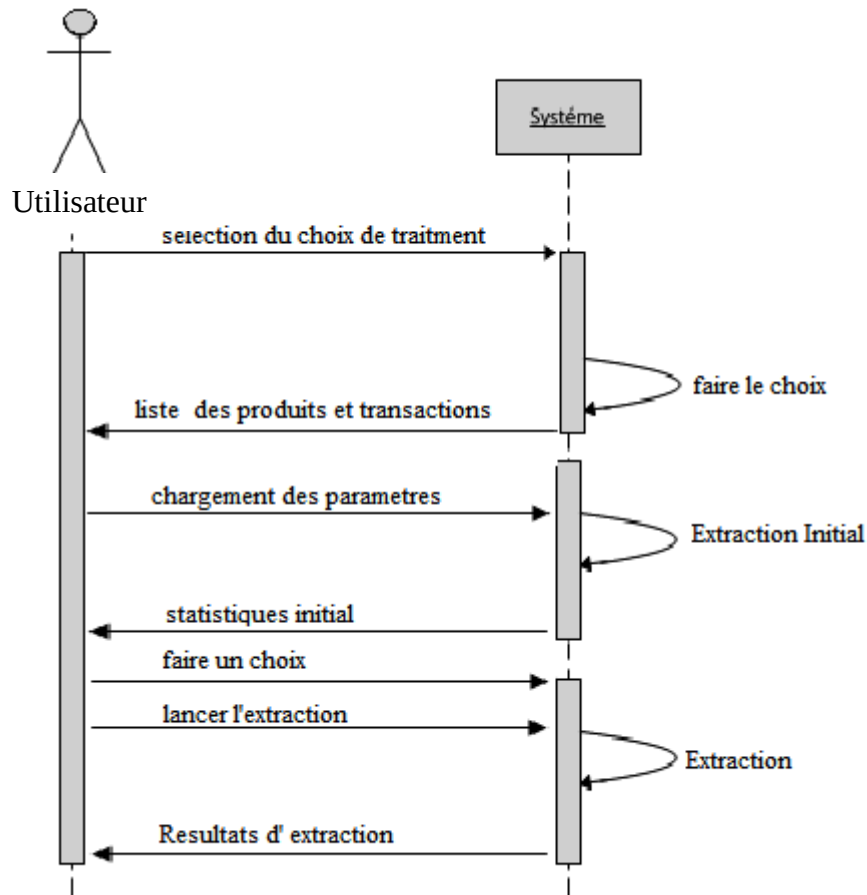


Figure IV.2. Diagramme de séquence de cas d'utilisation « Extraction ».

Description des scénarios

Scénarios de cas d'utilisation « Extraction »

- L' **Utilisateur** sélectionne le choix de traitement.
- Le système fait le choix et rend la liste des produits et transactions.
- L' **Utilisateur** charge les parametres.
- Le système fait une extraction et rend des statistiques initial .
- L' **Utilisateur** fait un choix .
- L' **Utilisateur** lance l'extraction.
- Le système fait l'extrait et rend le score d'extraction.

IV.2.2.4. Diagramme de séquence de cas d'utilisation « Résultats »

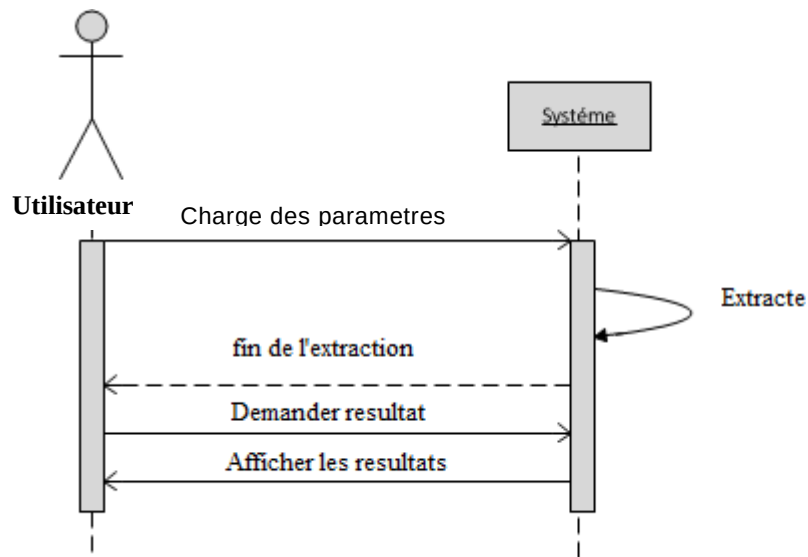


Figure IV.3. Diagramme de séquence de cas d'utilisation « Résultats ».

Description des scénarios

Scénarios de cas d'utilisation « résultats » :

- L'utilisateur initialise les paramètres.
- Le système extrait des résultats
- L'utilisateur demande les résultats
- Le système affiche les résultats.

IV.3. Mise en œuvre :

Après avoir conçu une solution qui, nous espérons, répondrait mieux aux objectifs fixés préalablement, nous entamons l'étape de réalisation de la plateforme logicielle de notre système.

Dans cette partie, nous mettrons en évidence les raisons de nos choix techniques (langage de programmation et les outils utilisés). Ensuite, nous allons citer les principales classes développées et décrire les différentes fenêtres de l'application.

IV.3.1. Outils de développement

Le choix du bon environnement de programmation est très important pour le développement des projets. Cela se fait suivant plusieurs facteurs : la puissance de compilation, la facilité d'utilisation, la disponibilité de plusieurs fonctionnalités, la communication avec d'autres environnements, etc.

Afin de réaliser notre système et son interface, nous avons eu recours aux outils : Microsoft Visual Studio 2008, .Net Framework.

Microsoft Visual Studio 2008 :

Microsoft Visual C# 2008 est un langage de programmation permettant la création d'une large gamme d'application s'exécutant sur le .Net Framework. C# est simple, puissant et orienté objet. Avec ses nombreuses innovations, C# permet le développement rapide d'applications tout en conservant la simplicité et l'élégance de style C.

La plateforme .Net contient une collection riche de classes et de bibliothèques prédéfinies qui facilitent l'implémentation des algorithmes.

Un Développement robuste qui permet la modification et le débogage efficace du code source. Possibilité d'exploiter les différentes DLL (Dynamic Link Library) créées par d'autres langages que le Visual C# sous la même plateforme.

IV.3.2. Description des différentes classes

L'ensemble des classes et les principales méthodes développées sont présentées dans le tableau suivant :

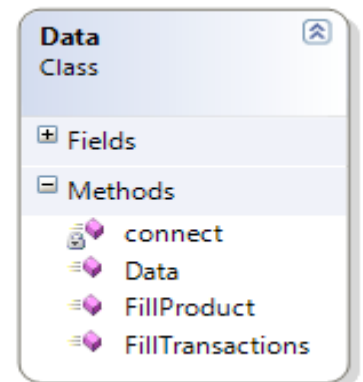
Classe	Signification
Data	Connecte et extrait les listes des transactions (ordonnances) et leur contenu du produits(médicaments).
Tools	Faire des différents recherches dans les listes .
result	Présente des statistiques initiaux.
Frequent	Trouve les listes des produits plus fréquentes .
Closed	Trouve les listes des produits plus fréquentes fermées .
Maximal	Trouve les listes des produits plus fréquentes maximal .
Strong	Génère l'ensemble des Règles d'associations .

TAB IV.2. Les principales classes de notre système.

• **Classe Data**

Cette classe permet de connecter notre système avec la base de données et extrait les listes des transactions (ordonnances) et leur contenu du produits(médicaments).

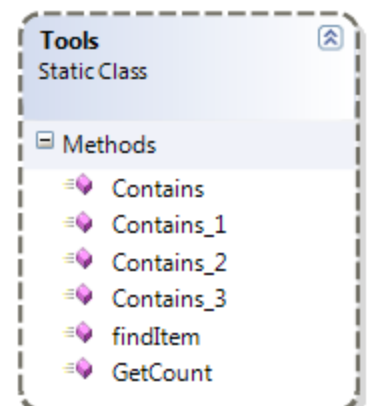
- **connect** : fait des requêtes à la BDD pour obtenir les transactions (ordonnances).
- **Data** : faire la connexion entre notre système et la base données.
- **FillTransactions** : défini les transactions dans des listes
- **FillProduct** : défini les produits des transactions dans des listes



• **Classe Tools**

Faire des différents recherches dans les listes avec les méthodes suivants :

- **Contains, Contains_1, Contains_2** : pour les recherches des listes dans des autres .
- **Contains_3** : recherche de produit dans tous les listes.
- **findItem** : recherche d'un produit dans une listes des



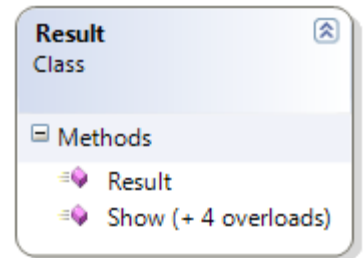
transactions .

- **GetCount** : calcule nombre d'occurrence d'un produit dans la listes des transactions

• Classe result

Présente des statistiques initiaux .

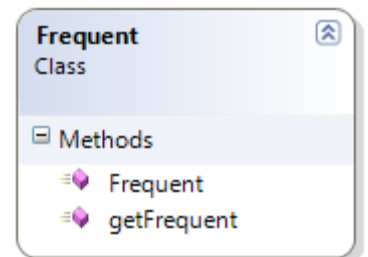
- **result** : calcule les nombres des items (produits) fréquentes ,fermées, maximal et règles .
- **show(+4 overloads)** : affiche les nombres des items (produits) fréquentes ,fermées, maximal et règles .



• Classe Frequent

Trouve les listes des produits plus fréquentes .

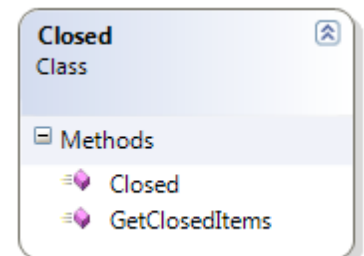
- **Frequent** : créer la liste des produits fréquentes.
- **getFrequent** : calcule et charge la liste des produits fréquentes.



• Classe Closed

Trouve les listes des produits fréquentes fermées .

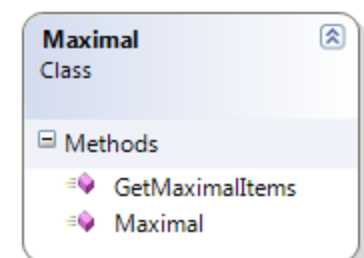
- **Closed** : créer la liste des produits fréquentes fermées .
- **getClosed** : calcule et charge la liste des produits fréquentes fermées .



• Classe Maximal

Trouve les listes des produits fréquentes maximal.

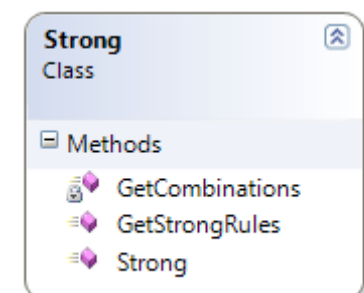
- **Maximal** : créer la liste des produits fréquentes maximal
- **GetMaximalItems** : calcule et charge la liste des produits fréquentes fermées .



• Classe Strong

Trouve les listes des règles d'associations et calcule leurs confiances .

- **Getcombinations** : combiner les listes des produits fréquentes maximaux



- **Strong** : créer la liste des règles .
- **GetStrongRules** : défini et charge la liste des regles

IV.3.3. Présentation de l'application

On présente dans cette section les différentes interfaces de l'application.

a) L'interface principale :



Figure IV.4. La forme principale.

- 1 Choix du traitement à tout les transactions de la BDD .
- 2 Choix du traitement à les transactions de la BDD dans un mois déterminé.
- 3 Choix du traitement à les transactions dans un mois et une année déterminées.
- 4 Choix du traitement à les transactions de la BDD dans une année déterminé.
- 5 Effectuer le traitement.
- 6 Fermer l'application .
- 7 Informations pour notre système .

b) Interface de traitement :

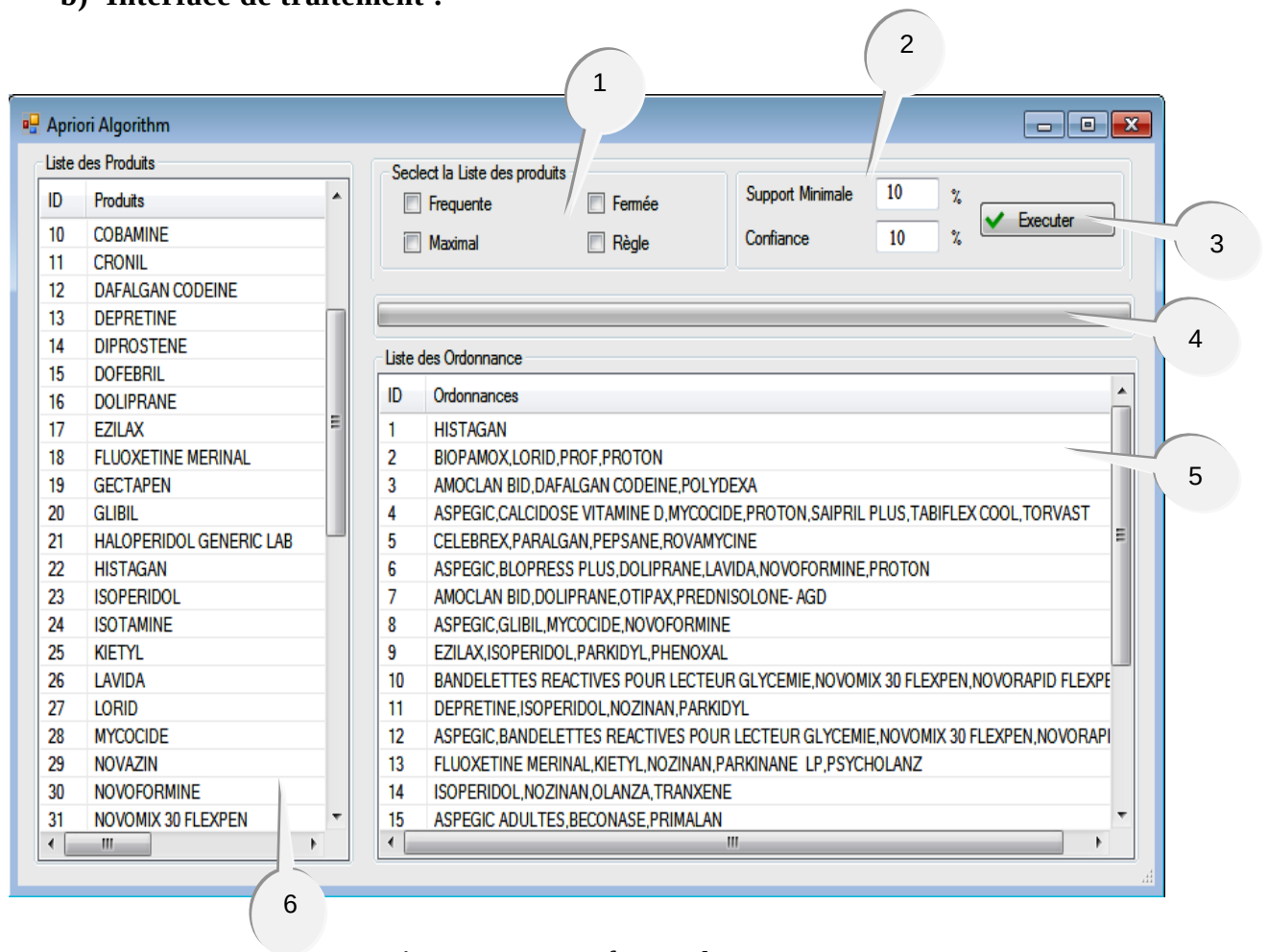
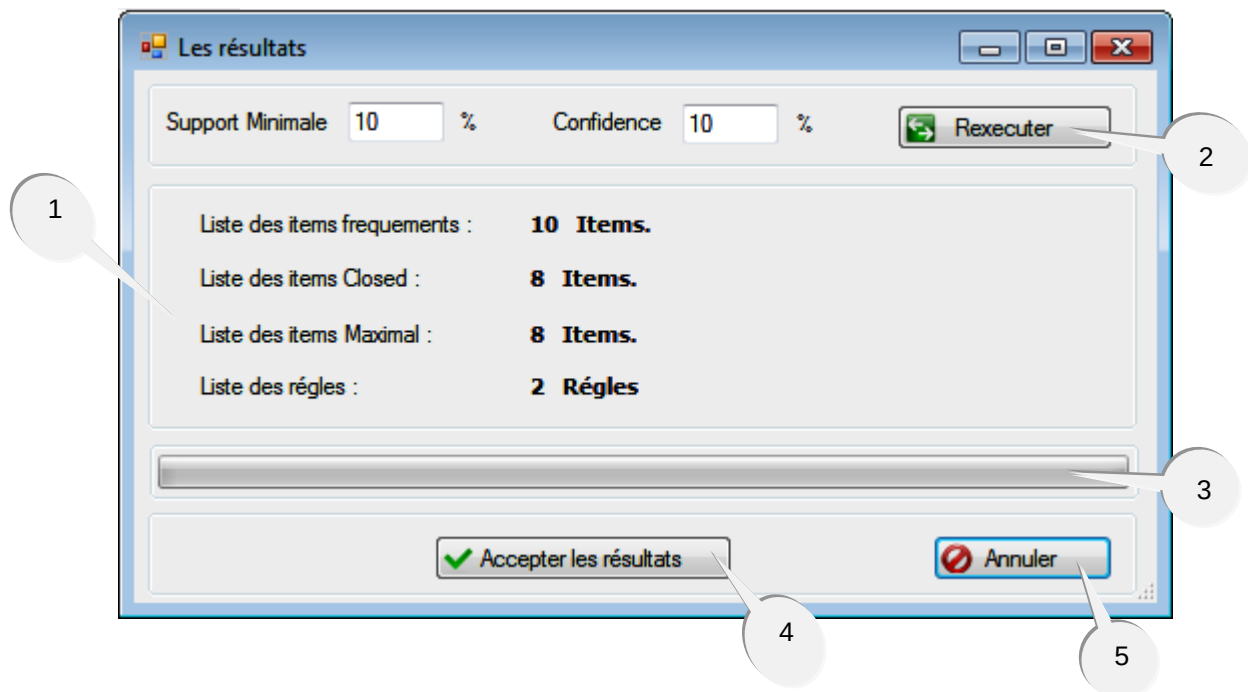


Figure IV.5. La forme de traitement.

- 1 Faire la sélection de liste (fréquentes ,fermées, maximal et règles) .
- 2 Saisir les critères de traitement (support et confiance minimal).
- 3 Exécuter l'application .
- 4 La barre de progression de traitement .
- 5 Liste des transactions (des ordonnances) .
- 6 Liste des produits (médicaments) .

c) Interface de statistiques :

**Figure IV.6.** La forme de statistiques .

1 Statistiques des listes des produits (fréquentes ,fermées, maximal et règles)

2 Réexécution après la modification des confiance et support minimal .

3 La barre de progression de traitement

4 Accepter les statistiques .

5 Annuler la modification des confiance et support minimal .

d) Les Interface des Résultats :

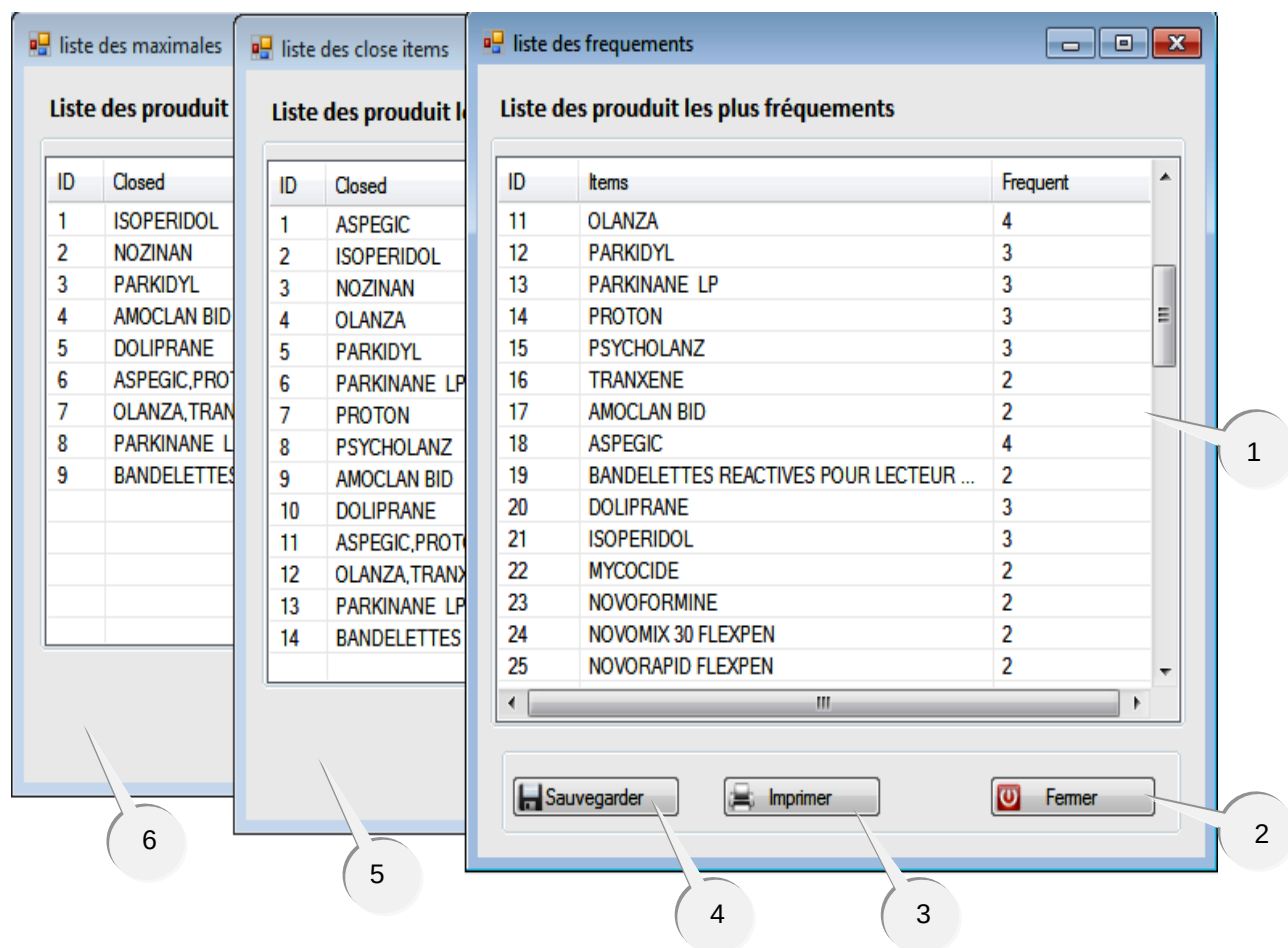


Figure IV.7 Interface des listes des produits (fréquentes ,fermées, maximal)

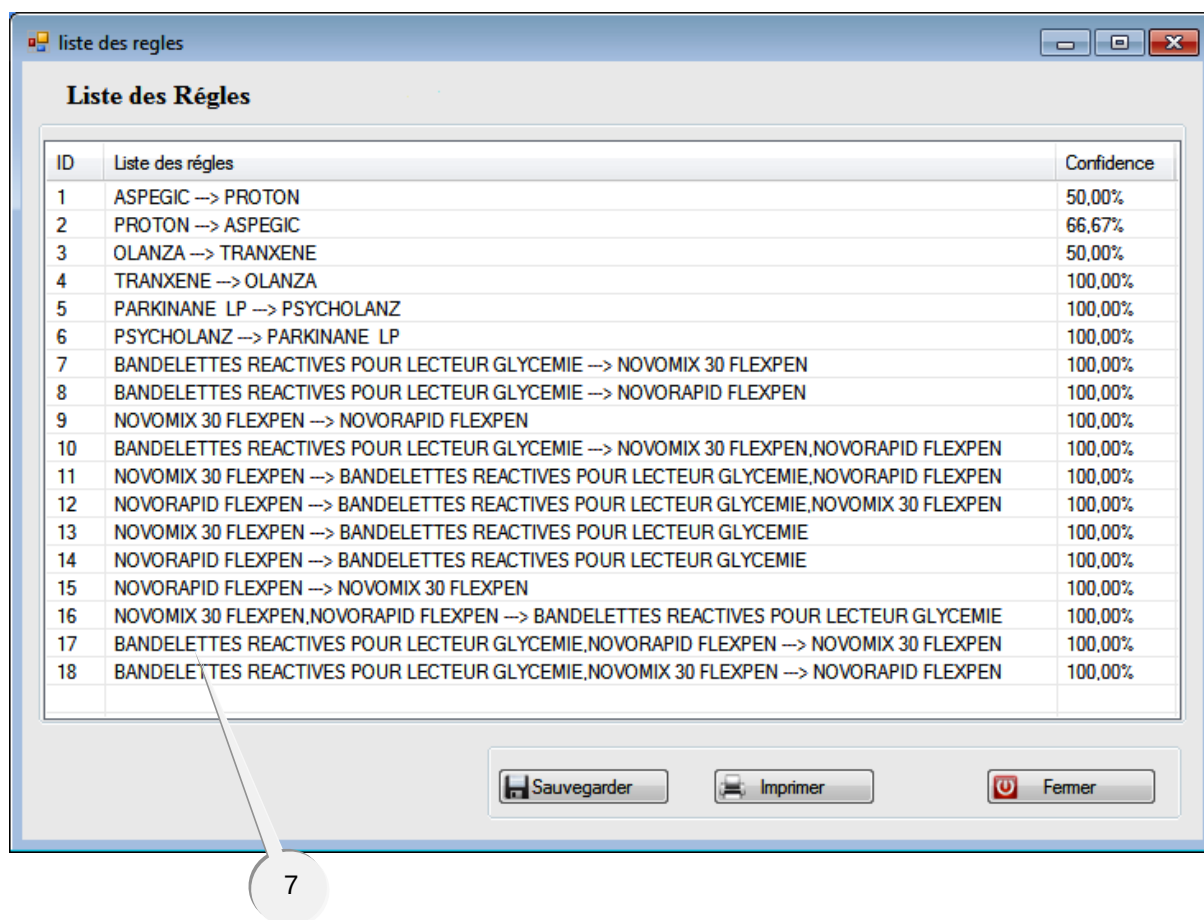


Figure IV.8 Interface de liste des règles .

- 1 liste des produits fréquentes.
- 2 Fermer cette interface
- 3 Imprimer cette liste .
- 4 Sauvegarder cette liste comme un fichier textuel .
- 5 liste des produits fréquentes fermées.
- 6 liste des produits fréquentes maximal.
- 7 liste des Règles .

IV.4. Conclusion :

Nous avons présenté dans ce chapitre l'étude conceptuelle de notre système en respectant la modélisation qui a été établie précédemment et ce, pour répondre, aux objectifs déclarés, ainsi nous avons détaillé l'implémentation en Visual C# en expliquant les principales classes que nous avons développées, et enfin nous avons présenté les différentes interfaces de notre système.

Conclusion générale

Nous avons présenté dans ce travail une approche assez récente de fouille de données qui est fondée sur la découverte de *règles d'association* à partir d'un ensemble de données qu'on appellera *transactions* (Agrawal et al. 1993). Ce thème est considéré aujourd'hui comme faisant parti des approches d'apprentissage symbolique non supervisé, utilisé dans le domaine de fouille de données (data mining) et d'extraction de connaissances.

Nous avons vu dans les chapitres. Le fouille de données et leurs méthodes d'une façon général.

Les règles d'association, étant une méthode d'apprentissage non supervisé, permettent de découvrir à partir d'un ensemble de transactions, un ensemble de règles qui exprime une possibilité d'association entre différents *items*. Une transaction est une succession d'items exprimée selon un ordre donné ; de même, l'ensemble de transactions contient des transactions de longueurs différentes. Nous avons présenté un exemple de l'utilité de cette approche est *l'ensemble des ordonnances* qui décrit un ensemble d'achats effectué à la *pharmacie* ; les règles d'association permet de découvrir de régularités dans l'ensemble de transactions comme par exemple : *Si PARAMOL alors ASPEGIC*, etc. Ces règles permettent par exemple au gérant de proposer des bons de réductions significatifs sur les achats futurs des clients et aident le pharmacien pour organiser et commander,Leurs produits ! !

Les plus importants résultats attendre par l'utilisation de cet algorithme sont les suivants :

➤ Au niveau organisationnel permet de cumuler (regrouper) les produits selon fréquemment achetés entre eux. , pour réduire le temps et la difficulté de du recherche.

- Assurer une meilleure gestion des commandes telles que le programme donne les importantes et les plus ventent ainsi le pourcentage de vente entre eux.
- Assurer une meilleur gestion des ressources monétaire ou bėjitaire , (réduire le coût).
- Permet de contrôler le coût de transport dans le cas où les produits sont stockés en des dépôts.
- Permet de faire une publicité à des produits peu ventes.
- Donne une vision au divers produit et leur corrélation par des autres facteurs (différents facteurs) pour la rendre plus lisible et pour permettre à l'utilisateur d'y appliquer quelques actions intéressantes.

Autrement dit, que notre système permet d'offrir à un gérant au niveau de la pharmacie, des grandes expériences dans une période très court.

Bibliographies

- **[Agr-93]** R. Agrawal and T. Imielinski and A. N. Swami (1993). *Mining Association Rules between Sets of Items in Large Databases* . Dans Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207_216.
- **[AgS-94]** R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In Proceedings of the 20th International Conference on VLDB, pages 487–499, Santiago, Chile, September 1994.
- **[Bla-05]** J. Blanchard. Un Système de Visualisation Pour l’Extraction, l’Évaluation, et l’Exploration interactives des Règles d’Association. Thèse, Sciences et Technologies de l’Information et des Matériaux, Nantes, November 2005. Henri Briand (Dir.).
- **[Bri-97]** S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic Itemset Counting and Implication Rules for Market Basket Data. In SIGMOD ’97 : Proceedings of the 1997 ACM SIGMOD international conference on Management of data, pages 255–264, New York, NY, USA, 1997. ACM.
- **[Bri-99]** R. J. Bayardo and R. Agrawal. Mining the Most Interesting Rules. In Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-99), pages 145–154. ACM Press, 1999.
- **[Fay-96]** Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors (1996). *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press.1, 5
- **[Gen-06]** L. Geng and H. J. Hamilton. Interestingness Measures for Data Mining : A Survey. *ACM Comput. Surv.*, 38(3) :9, 2006.

- **[Han-06]** Han ,Kamber,(2006) .Data Mining : Concepts and Techniques, 2nd ed. The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor.
- **[Han-02]** J. Han, J. Wang, Y. Lu, and P. Tzvetkov. Mining Top-K Frequent Closed Patterns without Minimum Support. In In Proceedings of ICDM02, pages 211–218, 2002
- **[Liu-99]** B. Liu,W. Hsu, and Y. Ma. Pruning and Summarizing the Discovered Associations. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD-99), pages 125–134. ACM Press, 1999.
- **[Liu-97]** B. Liu, W. Hsu, and S. Chen. Using General Impressions to Analyze Discovered Classification Rules. In KDD, pages 31–36, 1997.
- **[Pas-99]** N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient Mining of Association Rules Using Closed Itemset Lattices. Inf. Syst., 24(1) :25–46, 1999.
- **[Pas-07]**: Pascal Roques, Franck Vallée, “*UML 2 en action de l’analyse des besoins à la conception*“, 4^{ème} edition EROLLES 2007.
- **[Pia-94]** G. Piatetsky-Shapiro and C. J. Matheus. The Interestingness of Deviations. In Proceedings of KDD-94 workshop. AAAI Press, 1994.
- **[Sav-95]** A. Savasere, E. Omiecinski, and S. Navathe. An Efficient Algorithm for Mining Association Rules in Large Databases. In Proceedings of the 21st VLDB Conference, pages 432–443, Zurich, Switzerland, 1995.
- **[She-02]** Y.-D. Shen, Z. Zhang, and Q. Yang. Objective-Oriented Utility-Based Association Mining. In Proceedings of the 2002 IEEE International Conference on Data Mining, pages 426–433, 2002.
- **[Tan-04]** P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Objective Measure for Association Analysis. Information Systems, 29(4) :293–313, 2004.
- **[Toi-95]** H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hatonen, and H. Mannila. Pruning and Grouping Discovered Association Rules. In ECML’95MLnet workshop on statistics, machine learning, and knowledge discovery in databases, pages 47–52, April 1995.
- **[Tuf-07]** Stéphane Tufféry, *Data Mining et Statistique Décisionnelle*, Technip, nouvelle édition revue et enrichie, juin 2007

- **[Wan-05]** J. Wang, J. Han, Y. Lu, and P. Tzvetkov. TFP : An Efficient Algorithm for Mining Top-K Frequent Closed Itemsets. IEEE Transactions on Knowledge and Data Engineering, 17(5) :652–664, 2005.
- **[Zak-02]** M. J. Zaki and C.-J. Hsiao. CHARM:An Efficient Algorithmfor Closed Itemset Mining. In Proceedings of the Second SIAM DM, Arlington, VA, 2002. SIAM.
- **[1]** ([http:// www. zdnet. fr/ blogs/ 2005/ 11/ 27/ datamining/](http://www.zdnet.fr/blogs/2005/11/27/datamining/)) de ZDNet du 27 novembre 2005
- **[2]** : <http://jeff.cs.mcgill.ca/~godfried/teaching/projects97/azar/>, 2011