REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA
FACULTE DES MATHEMATIQUES ET DE L'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

MEMOIRE de fin d'étude

Présenté pour l'obtention du diplôme de MASTER

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Systèmes d'Informations Avancés

Par: Layadi Kanza

SUJET

Extraction des motifs séquentiels

Soutenu publiquement le : 01 / 06 /2016 devant le jury composé de :

A. Khettaf	**Université de M'sila**	**Président**
S. Guesmia	**Université de M'sila**	**Rapporteur**
A. Bouda	**Université de M'sila**	**Examinateur**

Promotion : 2015 /20 16

Remerciement

Je tiens à exprimer mes vifs remerciements à ALLAH qui je
A donné la patience afin de finir mon étude malgré les
Difficultés rencontrées.
Je souhaite d'abord exprimer ma vive reconnaissance pour mon encadreur
Monsieur SALAH GUASMIA qui a bien voulu
Accepter l'encadrement de ce mémoire et il m'a toujours guidé sans jamais me contraindre,
Je le remercie très sincèrement pour m'avoir donné l'opportunité de mener à bien ce
Travail par sa disponibilité et son soutien, pour le temps qu'il m'a consacré et ses nombreux et précieux conseils.
Je souhaite également remercier les membres du jury pour j'avoir fait l'honneur d'accepter de juger et d'évaluer mon travail.
Encore je remercier ma famille est mes amies pour leurs encouragements.

Table Des Matières

Liste des tableaux ……. 1

Liste des figures …… 2

INTRODUCTION GENERALE …… 3

CHAPITRE 01 : GENERALISATION SUR LA FOUILLE DE DONNEE

1. Introduction…….. 5

2. Extraction de connaissance a partir de données ECD……..... 5

2.1.La sélection ……..... 6

2.2.Prétraitement et transformation desdonnées…… 6

2.2.1. Le prétraitement et le nettoyage des données ……. 6

2.2.2. La transformation des données ……...……... 6

2.3. La fouille de données ……..... 6

2.4.L'interprétation et l'évaluation des informations ……..…… 6

3. Définition de la fouille de donnée……... 7

4. Les types de données qui sont appliqués par la fouille de données……..... 8

5. Tâches de fouille de données …… 8

5.1.La classification ……...…… 9

5.2.L'estimation…….. 9

5.3.La prédiction ……. 10

5.4.Le clustering ……...……. 10

5.5.L'association……...……. 10

5.6.La description …….. 11

6. Les techniques de fouille de donnée……. 11

6.1.Les techniques prédictives (apprentissage supervisé) …… 11

6.1.1. L'arbre de décision……...……..... 11

6.1.2. Les réseaux de neurones ……. 12

6.1.3. L'algorithme des k-Plus proches voisins ……... 13

6.2.Les techniques descriptives (apprentissage non supervisé)……..... 14

6.2.1. Clustering (segmentation) ……..... 14

6.2.1.1.L'algorithme de k-moyennes (k means)……..……. 14

6.2.2. Les règles associatives …..…..……. 14

6.2.3. Les motifs séquentiels (en anglais sequencemining)……..... 15

7. Domaines d'application …….. 15

8. Conclusion ……..... 17

CHAPITRE 02 : LES REGLES D'ASSOCIATION

1. Introduction ……..……. 19
2. Les règles d'association ……..... 19
   2.1.Domain d'application les règles d'association ……..……. 20
   2.2.Les étapes d'extraction des règles d'association…….…….... 21
      2.2.1. Sélection et préparation des données ……... 22
      2.2.2. Découverte des itemsets fréquents ……..……...…. 22
      2.2.3. Génération des règles d'association ……..……..….. 23
      2.2.4. Visualisation et interprétation des règles d'associations …….... 23
3. Concepts généraux ……... 23
   3.1.Définition …….……...…. 23
4. L'extraction les règles d'association ……..... 26
   4.1.extraction des Itemsets fréquents …… 26
      4.1.1. Approche d'extraction des itemsets fréquents ……...…..…… 27
      4.1.2. Approche d'extraction d'itemsets maximaux ……...…….. 27
      4.1.3. Approche d'extraction d'itemsets fermés fréquents ……..... 28
   4.2.génération des règles d'association ……..... 28
5. Algorithme d'extraction des règles d'association ……..……..... 28
   5.1.Algorithme Apriori ……...…….... 28
   5.2.Algorithme FP-growth ……...…….. 31
   5.3.Algorithme Eclat …….……..... 31
   5.4.L'algorithme partition ……...…….…. 32
6. Mesures de qualité des règles d'association ……...……. 32
7. Classification des algorithmes ……...……... 34
   7.1. Stratégie de Calcul du Support ……..……... 34
   7.2.Direction de Recherche …….…… 35
   7.3.Stratégie de recherche ……..…….. 35
8. Conclusion …… 37

CHAPITRE 03 : LES MOTIFS SEQUENTIELS

1. Introduction ……..……. 39
2. motifs séquentiels ……..……... 39
   2.1. Exemples d'applications possibles ……...…. 39
   2.2.Extraction de motifs séquentiels versus extraction d'itemsets et de règles d'association ……. 40

3. Concepts généraux ……. 41
3.1. Définitions ……..... 41
3.2.Propriétés des séquences fréquentes ……..…….. 43
4. Extraction les motifs séquentiels ……..… 43
4.1.Méthodes horizontales …….…… 43
4.1.1. La méthode GSP (GeneralizedSequential Patterns) ……...……. 43
4.1.1.1. Limites de l'algorithme GSP ……..…… 45
4.1.2. L'algorithme PSP (Prefix Tree for Sequential Pattern)…….. 46
4.1.2.1. Limites de l'algorithme PSP ……..…. 47
4.2.Méthode verticale ……..... 47
4.2.1. L'algorithme SPADE …….. 47
4.2.1.1. Limite de SPADE ……... 49
4.3.Méthode par projection ……..... 49
4.3.1. L'algorithme FreeSpan ……..... 50
4.3.2. L'algorithme PrefixSpan ……..... 50
5. Conclusion ……..... 52
CHAPITRE 4 : Implémentation
1. Introduction …….. 54
2. Implémentation …… 54
2.1. La fonctionnement de l'algorithme Generalized Sequential Pattern (GSP) :……..…….. 54
2.2. La fonctionnement de l'algorithme SPADE …… 56
3. Environnement de l'application ……. 60
4. L'architecture de l'application …… 62
5. Conclusion ……. 63
CONCLUSION GENERALE 64
Bibliographie

Liste des Tableaux

Table 2.1 : Représentation binaire des données de paniers de clients

Table 3.1 : Exemple base de données de séquences

Table 3.2 : base de données exemple pour PrefixSpan.

Table 3.3 : Résultat de PrefixSpan sur la base de données de la table 3.2

Table 4.1. Une base de données de séquence.

Table 4.2. Les candidats de 1-séquence.

Table 4.3. Les 1-séquence fréquents par GSP

Table 4.4. Les candidats de 2-séquence par GSP

Table 4.5. les 2-séquence fréquents (GSP).

Table 4.6. un 3-séquence fréquents(GSP)

Table 4.7. un 4-séquence fréquent.(GSP)

Table 4.8. ID_List des items fréquent.

Table 4.9. Les 1-séquence fréquent (spade).

Table 4.10 id-liste 1-séquence temporelle

Table 4.11. id-liste 1-séquence non temporelle.

Table 4.11. les 2-séquence fréquent(SPADE).

Table 4.12. id-liste2-séquence temporelle.

Table 4.13. id-liste 2-séquence non temporelle.

Table 4.14. les 3-séquence résultant par (SPADE)

Table 4.15. id liste de 4-séquence

Table 4.16. id-liste de 4-séquence

Liste des Figures

Figure 1.1 : les processus général de KDD

Figure 1.2 : Exemple d'arbre de décision

Figure 1.3 : Vue simplifiée d'un réseau artificiel de neurones

Figure 2.1 : Les étapes d'extraction les règles d'association

Figure 2.2 : Exemple de treillis associé à un ensemble de 5 items

Figure 2.3 : Pseudo code de l'algorithme Apriori

Figure 2.4 : Pseudo code de l'algorithme Apriori-Gen

Figure 2.5 : Recherche en largeur

Figure2.6 : Recherche en profondeur

Figure 2.7 : Classification des algorithmes d'extraction des règles d'association

Figure 3.1 : diagramme représente l'algorithme GSP

Figure 3.2 : La structure de données utilisée par l'algorithme GSP

Figure 3.3 : l'arbre de préfix PSP et l'arbre de GSP

Figure 3.4 : classes d'équivalences générer par l'algorithme SPADE

Figure 3.5 : Le pseudo code de l'algorithme SPADE

Figure 4.1 : L'architecture de l'application

Figure 4.2 : interface graphique de l'application

INTRODUCTION GENERALE

Avec le développement des outils informatiques, nous assistons ces dernières années à un accroissement considérable de la quantité d'informations stockées dans de grandes bases de données scientifiques, économiques, financières, médicales, etc. et le défi aujourd'hui n'est plus de stocker ces donnée mais d'en extraire de l'information implicite et cachée dans ces données, particulièrement des recherches sur l'**E**xtraction automatique de **C**onnaissances à partir de **D**onnées. Cette discipline est l'intersection des domaines des bases de données, l'intelligence artificielle, et la statistique. L'ECD est décrite comme un processus interactif d'extraction de connaissances à l'aide d'algorithmes de calcul et d'interprétation des résultats, lors d'interactions avec l'expert pour aider à la décision, à partir d'ensemble de méthodes statistiques et algorithmiques sous la terminologie de Data Mining (la fouille de données).

La fouille de données concerne l'étape algorithmiquement difficile de ce processus, qui produit des motifs potentiellement intéressants à partir des données, elle regroupe un certain nombre de tâches, telles que la prédiction, le regroupement par similitude, la classification, la découverte d'associations, etc. L'un des plus importants problèmes de la fouille de données est la recherche de règles d'association. Cette approche, spécialisé dans la gestion de la relation client (GRC) et elle est identifier des corrélations cachées, potentiellement utiles, entre les attributs d'une base de données, il y a plusieurs approches et algorithmes ont été élaborés afin d'extraire les motifs et les règles d'association. Plusieurs types de motifs ont été définis selon le type de corrélation à extraire et selon la nature des données. Et dans ce mémoire on a plus précisément parlé sur les algorithmes d'extraction des motifs séquentiels (des motifs temporels) pour la découverte d'enchaînements fréquents dans les bases de données, avec des contraintes temporelles et l'identification des événements d'individus afin de pouvoir suivre leurs comportements séquentiels au cours du temps.

L'objectif de ce projet est la compréhension du comportement des principaux algorithmes d'extraction de motifs séquentiels en expliquant et illustrant leur fonctionnement l'implémentation des algorithmes pour l'extraction des règles séquentiels. Ce mémoire est structuré en quatre chapitres :

Le premier chapitre comporte une description générale sur la fouille de données et leurs techniques.

Le deuxième chapitre détermine les différents concepts d'extraction des règles d'association et mentionner les algorithmes les plus pratique.

Le troisième chapitre est consacré à quelques algorithmes d'extraction des motifs séquentiels.

Enfin, le quatrième chapitre est consacré à l'environnement logiciel et matériel utilisé ainsi que l'implémentation d'un algorithme.

CHAPITRE 01

GENERALISATION SUR LA FOUILLE DE DONNEES

1. Introduction :

De nos jours, les changements de notre environnement sont dénotés par des capteurs qui sont devenus de plus en plus nombreux. Par conséquent, la compréhension de ces données est très importante. Et comme il est dit Piatestky-Shapiro, « […] as long as the world keeps producing data of all kinds […] at an ever increasing rate, the demand for data mining will continue to grow» [1]. D'où la fouille de données devient une nécessité, En 1989, Gregory Piatetsky-Shapiro a donné un nom à une discipline qui elle peut faire la fouille de donnée : Extraction de Connaissances à partir de Données (ECD), ou en Anglais, Knowledge Discovery in Databases (KDD).[2]

Dans ce chapitre, nous commençons par présenter le processus d'extraction de connaissances (ECD) qui constitue le cadre général dont lequel s'inscrit notre travail. Nous passons en revue à l'étape de la fouille de données, leurs taches, leurs techniques et puis leurs domaines d'application.

2. Extraction de connaissance à partir de données ECD :

L'ECD est un processus semi-automatique et itératif, constitué de plusieurs étapes allant de la sélection et la préparation des données (prétraitement) jusqu'à l'interprétation et l'évaluation des données (post-traitement), en passant par la phase de recherche d'informations (la fouille des données). Ces quatre phases sont illustrées dans la figure ci dessous et développées dans ce qui suit:[3]

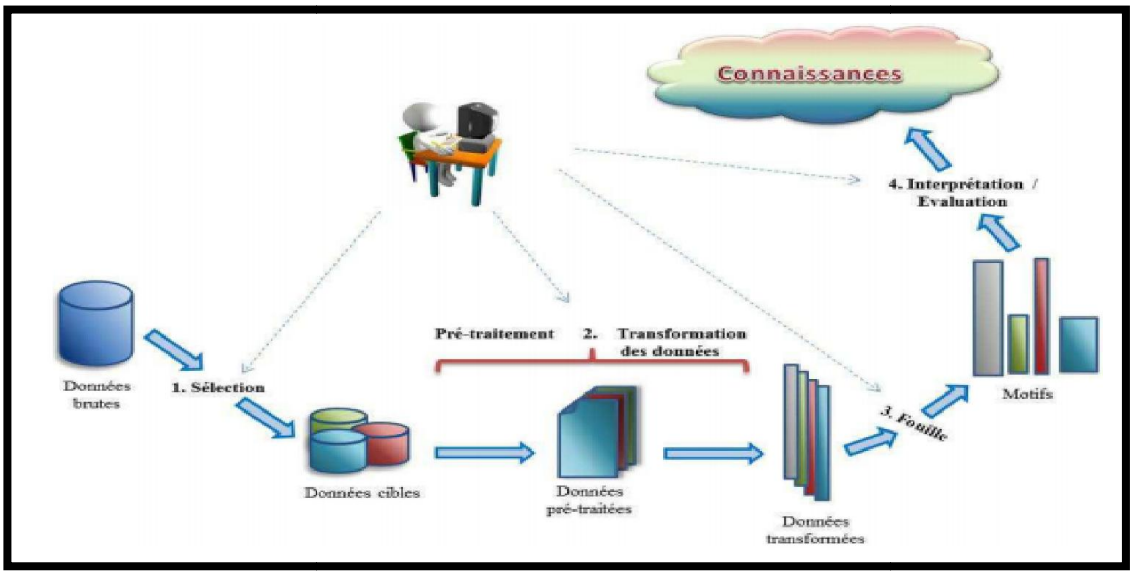


Figure 1.1 : les processus général de KDD.

2.1. La sélection :

Cette première étape du processus consiste à garder parmi l'ensemble des données, qu'une sous-partie intéressante par rapport au problème. Ainsi, il s'agirait en effet d'identifier les informations adaptées aux besoins de l'utilisateur pour une application bien déterminée.

2.2. Prétraitement et transformation des données: elle divisé a deux sous étapes qui sont :

2.2.1. Le prétraitement et le nettoyage des données :

Cette étape inclut des opérations comme l'enlèvement du bruit et des valeurs aberrantes -si nécessaire, des décisions sur les stratégies qui vont être utilisées pour traiter les valeurs manquantes,

2.2.2. La transformation des données :

Cette étape est très importante pour la réussite du projet et doit être adaptée en fonction de chaque base de données et des objectifs du projet. Dans cette étape nous cherchons les méthodes correctes pour représenter les données. Ces méthodes incluent la réduction des dimensions et la transformation des attributs. Une fois que toutes ces étapes seront terminées, les étapes suivantes seront liées à la partie de Data Mining, avec une orientation sur l'aspecte algorithmique. [4]

2.3. La fouille de donnée:

Cette troisième étape est au cœur du processus de l'ECD puisqu'elle permet d'identifier et de mettre en évidence des informations ou des connaissances à partir de données transformées. Les informations générées peuvent prendre différentes formes selon la méthode utilisée et le problème à résoudre : tels que les arbres de décisions (l'analyse prédictive) ou les règles d'association (l'analyse exploratoire).

2.4. L'interprétation et l'évaluation des informations :

Une quantité importante d'informations peut être générée à partir des algorithmes d'extraction des données, dont la plupart sont inutiles ou redondantes. Pour pallier ce défaut, une dernière étape de post-traitement des connaissances découvertes s'avère indispensable pour transformer ces connaissances extraites en connaissances intéressantes et facilement exploitables par l'utilisateur. Il serait donc intéressant d'appliquer un filtrage automatique sur l'ensemble des informations extraites et de les ordonner afin de faciliter la prise de décision et l'interprétation des résultats révélés à l'utilisateur. Dans la littérature, il existe plusieurs

moyens pour assister l'utilisateur dans son travail, comme l'utilisation de mesures d'intérêt qui permettraient d'évaluer la qualité des connaissances extraites.

3. Définition de fouille de donnée :

Plusieurs définitions ont été proposées dans [5], la fouille de donnée serait:

- " la découverte de nouvelles corrélations, tendances et modèles par le tamisage d'un grand nombre de données ";
- " un processus d'aide à la décision où les utilisateurs cherchent des modèles d'interprétation dans les données" ;
- " l'extraction d'informations originales, auparavant inconnues, potentiellement utiles à partir des données " ;
- un processus de support de décision à la recherche de motifs imprévus et inconnus dans de grandes bases de données, Parsaye - Friedman(1997) ;[2]
- un processus de découverte de motifs avantageux dans les données John (1997) ;[2]

  D'après Han et Kamber [6] [7], le terme datamining se réfère à l'extraction de connaissances à partir de grandes quantités de données. La fouille de donnée est un domaine récent qui se situe à l'intersection des statistiques, de l'apprentissage automatique et des bases de données.
- D'après [8], la définition la plus communément admise de fouille de donnée est: «La fouille de donnée est un processus non trivial qui consiste à identifier, dans des données, des schémas nouveaux, valides, potentiellement utiles et surtout compréhensibles et utilisables».

4. Les types de données qui sont appliqués par la fouille de données :

En principe, la fouille de données peut s'appliquer à tous les types de données. Toutefois, selon chaque type de données, les algorithmes de la fouille de données diffèrent. Quelques exemples de types de données (Zaïane (1999)) auxquels peut s'appliquer la fouille de données sont [4] :

"Flat file" les fichiers plats: sont actuellement la source de donnée le plus commune pour les algorithmes du fouille de donnée, ils sont des fichiers en format texte ou binaire, contenant un enregistrement par ligne, avec des champs séparés par des délimiteurs, tels que les virgules ou les tabulations. Dans ce type de fichiers, les données peuvent être des transactions, des séries temporelles etc .

Base de données relationnelle : est une base de données consistant dans des tableaux séparés, avec des liaisons explicitement dénies et dont les éléments peuvent être combinés sélectivement comme des résultats à des interrogations. Chaque tableau contient des colonnes (correspondantes à des tuples) et des lignes (correspondantes à des attributs), La fouille de donnée peut profiter du SQL pour la sélection, la transformation et la consolidation;

Les data warehouses: est un support de données dans laquelle est centralisé un volume important de données consolidées à partir des différentes sources de données (souvent hétérogènes), par exemple Si le directeur de l'entreprise veut accéder aux données de tous les magasins pour prendre des décisions stratégiques, il serait plus approprié si toutes les données étaient stockées dans un seul emplacement avec une structure homogène qui permet l'analyse interactive des données.
Autrement dit, les données de différents magasins peuvent être chargées, nettoyées, transformées et intégrées ensemble. Pour faciliter la prise de décisions et les vues multidimensionnelles ;

Base de données transactionnelle : est un ensemble d'enregistrements représentant des transactions, Une transaction contient un identifient unique (transactionID) et une liste d'items composant la transaction.[7]

Bases de données orientées objet et relationnelle objet : Il s'agit d'un type spécial de base de données (ou base de données relationnelle) où les données sont des objets [2].

Les bases de données multimédia : comportent des documents sonores, des vidéos, des images et des médias en textes et audio. Elles peuvent être stockées sur des bases de données orientées objets ou objets relationnelles ou simplement sur un fichier système. Le multimédia est caractérisé par sa haute dimension ce qui rend le datamining sur ce type de données très difficile.[9]

5. Tâches de fouille de données :

Après avoir défini ce qu'est la fouille de donnée, il convient de présenter les tâches que celui-ci peut effectuer. Beaucoup de problèmes d'intérêt commercial intellectuel, économique, et peuvent être exprimés en termes de six tâches suivantes [10] :

- La classification
- L'estimation
- La prédiction
- Le clustering
- L'association
- La description

5.1. La classification :

Etant donné un ensemble prédéfini de classes d'objets, affecter un objet à une classe, selon une certaine mesure de proximité est le rôle de la classification.

La classification consiste à mettre à jour chaque enregistrement en déterminant un champ de classe, les techniques de classification commencent par définir un plan d'expérience ou un ensemble de données d'apprentissage sur lequel on applique les méthodes de classification. L'objectif est de créer un modèle qui peut être appliqué aux données non classifiées dans le but de les classifiées [9]

Quelques exemples de l'utilisation des tâches de classification dans les domaines de recherche et commerce sont les suivants : [11]

- attribuer ou non un prêt à un client.
- accepter ou refuser un retrait dans un distributeur.
- Diagnostiquant si une certaine maladie est présente.
- Déterminer quels numéros de téléphone correspondent aux fax.[9]

5.2. L'estimation :

L'estimation est similaire à la classification excepté que la variable cible est continue au lieu d'être catégorielle [12]. La classification se rapporte à des événements discrets (par exemple : le patient à été ou non hospitalisé). L'estimation, elle porte sur des variables continues (par exemple: la durée d'hospitalisation)[13]. Par exemple on cherche à estimer La lecture de tension systolique d'un patient dans un hôpital, en se basant sur l'âge du patient, son genre, son indice de masse corporelle et le niveau de sodium dans son sang. La relation entre la tension systolique et les autres données vont fournir un modèle d'estimation. Et par la suite nous pouvons appliquer ce modèle dans d'autres cas[9].

Quelques exemples de l'utilisation des tâches d'estimation dans les domaines de recherche et commerce sont les suivants :

- Estimer le nombre d'enfants dans une famille [9].
- Estimant le montant d'argent qu'une famille de quatre membres choisis aléatoirement dépensera pour la rentrée scolaire.
- Estimer les revenus d'un client.[11]

5.3. La prédiction :

Elle est similaire à la classification et à l'estimation mise à part que pour la prévision. Les résultats portent sur le futur [12]. La seule méthode pour mesurer la qualité de la prédiction est d'attendre. Quelques exemples de l'utilisation des tâches de prédiction dans les domaines de recherche et commerce sont les suivants [4] :

- Prévoir les gagnants du championnat de football en se basant sur une comparaison des résultats des équipes.
- Prévoir quels clients va déménager dans les 6 mois qui suivent.

5.4. Le clustering :

Le clustering (segmentation) porte sur le regroupement d'enregistrements, d'observations ou de cas en groupe d'objets similaires. Il est différent de la classification parce qu'ils n'y a pas de variable cible pour segmenter. Le clustering a pour objectif de segmenter l'ensemble entier des données en des sous groupes relativement homogènes, des clusters, dans lesquels la similarité des enregistrements dans le groupe est maximisée et la similarité en dehors du groupe est minimisée[12].

Les algorithmes du clustering peuvent être appliqués dans des différents domaines, tel que :

- Découvrir des groupes de clients ayants des comportements semblables. [9]
- segmentation des plantes et des animaux étant donné leurs caractéristiques.

5.5. L'association :

Cette fonction du fouille de donnée permet de découvrir quelles variables vont ensemble, quelles sont les règles qui vont permettre de quantifier les relations entre deux ou plusieurs variables. [4]

Elle extrait les corrélations entre les données. Très répandue dans le secteur de la distribution car leur principale application est « l'analyse du panier de la ménagère » qui consiste en la recherche d'associations entre produits sur les tickets de caisse. Trouver tous les articles achetés ensemble et ceux qui ne sont jamais achetés ensemble

dans un supermarché est un exemple de la fonction d'association. [7].Les règles d'associations sont de la forme "Si antécédent, alors conséquent".

Quelques exemples de l'utilisation des tâches des réglés d'associations dans les domaines de recherche et commerce sont les suivants: [9]

- Trouver dans un supermarché quels produits sont achetés ensemble et quels sont ceux qui ne s'achètent jamais ensemble.
- Déterminer la proportion des cas dans lesquels un nouveau médicament peut générer des effets dangereux.

5.6. La description :

Parfois le but du fouille de donnée est simplement de décrire se qui se passe sur une Base de Données compliquée en expliquant les relations existantes dans les données pour en premier lieu comprendre le mieux possible les individus, les produit et les processus présents sur cette base. Une bonne description d'un comportement implique souvent une bonne explication de celui-ci [9].

6. Les techniques de fouille de donnée :

Sous ce vocable sont regroupées les méthodes mathématique et algorithmique permettant d'effectuer les taches de fouille de donnée. On peut citer :

6.1. Les techniques prédictives (apprentissage supervisé) :

Est une technique d'apprentissage automatique plus connu sous le terme anglais de machine learning qui permet à une machine d'apprendre à réaliser des tâches à partir d'une base d'apprentissage contenant des exemples déjà traités. Chaque élément (item) de l'ensemble d'apprentissage (training set) étant un couple entrée-sortie.

De part sa nature, l'apprentissage supervisé concerne essentiellement les méthodes de classification de données (on connaît l'entrée et l'on veut déterminer la sortie) et de régression (on connaît la sortie et l'on veut retrouver l'entrée). [15]

6.1.1. L'arbre de décision:

Est un outil puissant et populaire pour la classification et la prédiction. Il permet la représentation graphique d'une procédure de classification et il a une traduction immédiate en termes de règles de décision [15].

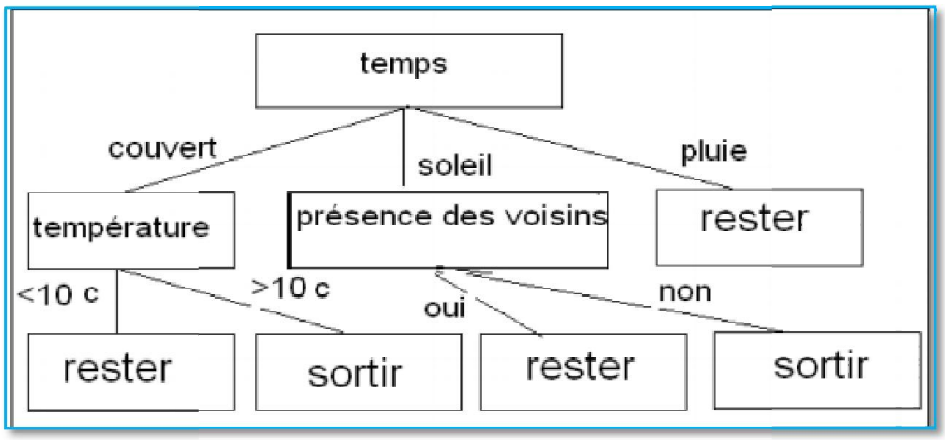


Figure 1.2. Exemple d'arbre de décision.[16]

La construction de l'arbre de décision se fait de façon récursive, en découpant successivement l'ensemble d'exemples comme suit : [16]

1. Si tous les exemples forment une seule classe, alors créer une feuille pour cette classe ;
2. Sinon, choisir le meilleur sélecteur (un attribut) qui représentera un nœud, puis :
   - Fixer un test (le plus discriminant possible) sur cet attribut ;
   - Découper l'ensemble d'exemples suivant ce test selon les valeurs possibles de cet attribut, ce qui représente les branches du nœud ;
   - Pour chaque nouvel ensemble, construire un sous arbre de décision (refaire les étapes 1 et 2).

Les algorithmes de construction d'arbre de décision procèdent souvent par deux phases qui sont : [4]

1. Construction d'un arbre par division récursive des nœuds.
2. Elagage de l'arbre récursivement depuis les feuilles afin de réduire sa taille.

6.1.2. Les réseaux de neurones:

«Les réseaux de neurones sont des outils très utilisés pour la classification, l'estimation, la prédiction et la segmentation. Ils sont issus de modèles biologiques, sont constitués d'unités élémentaires (les neurones) organisées selon une architecture» [11].

Les réseaux de neurones constituent l'un des techniques les plus récentes permettant d'induire un ensemble de valeurs en sortie à partir d'un ensemble de valeur en entrée. Il est constitué d'un grand nombre de cellules interconnectées et d'une succession de couches dont chacune prend ses entrées sur les sorties de la précédente. Chaque couche (i) est

composée de N_i neurones, prenant leurs entrées sur les N_{i-1} neurones de la couche précédente [5].

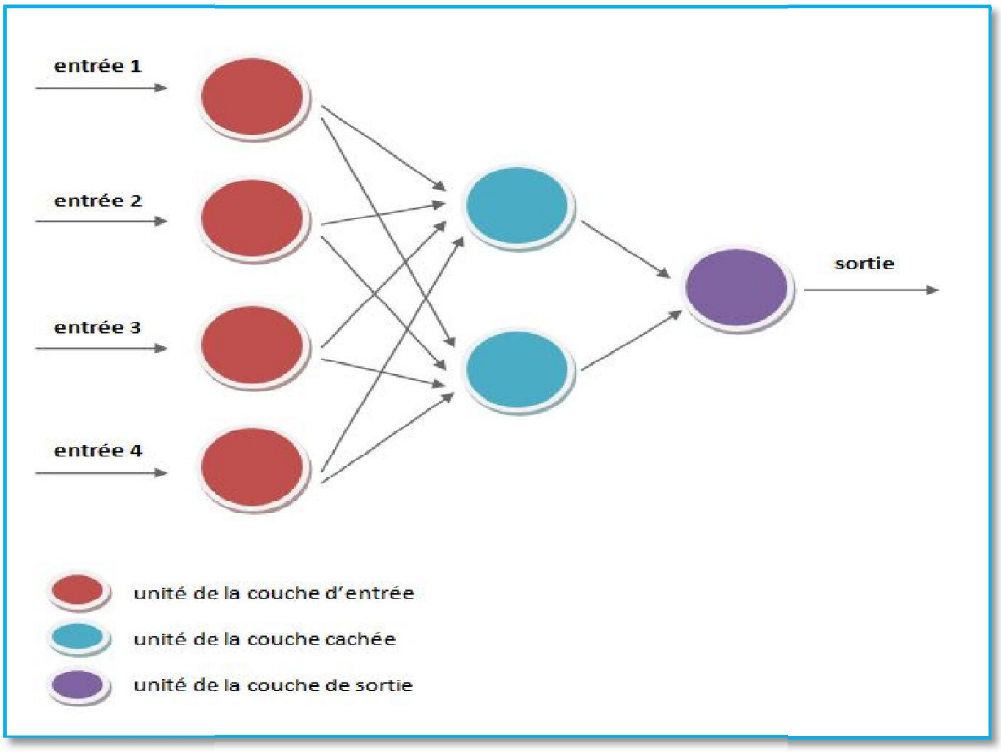


Figure 1.3. Vue simplifiée d'un réseau de neurones.

6.1.3. L'algorithme des k-Plus proches voisins:

La méthode des plus proches voisins (PPV, k nearest neighbor en anglais ou KNN) est une algorithme dédiée à la classification qui peut être étendue à des tâches d'estimation. Est une méthode de raisonnement à partir de cas. Elle part de l'idée de prendre des décisions en recherchant un ou des cas similaires déjà résolus en mémoire. Le but de cet algorithme est de prendre des décisions en se basant sur un ou plusieurs cas similaires déjà résolus en mémoire Dans ce cadre, et Contrairement aux autres méthodes de classification (arbres de décision, réseaux de neurones, …etc.) l'algorithme de KNN ne construit pas de modèle à partir d'un échantillon d'apprentissage, mais c'est l'échantillon d'apprentissage, la fonction de distance et la fonction de choix de la classe en fonction des classes des voisins les plus proches, qui constituent le modèle [16].

Algorithme de classification par k-PPV :

Paramètre : le nombre k de voisins

Donnée : un échantillon de m exemples et leurs classes

La classe d'un exemple X est c(X)

Entrée : un enregistrement Y

1. Déterminer les k plus proches exemples de Y en calculant les distances
2. Combiner les classes de ces k exemples en une classe c

Sortie : la classe de Y est c(Y)=c [18].

6.2. Les techniques descriptives (apprentissage non supervisé) :

Lorsque l'on cherche à extraire des informations nouvelles et originales d'un ensemble de données dont aucun attribut n'est plus important qu'un autre.

Le résultat des algorithmes de fouille de donnée non supervisé doit être analysé afin d'être retenu pour un usage ou tout simplement rejeté [14].

6.2.1. Clustering (segmentation) :

Est une méthode statistique d'analyse de données qui a pour but de regrouper un ensemble de données en différents groupes homogènes. Chaque sous-ensemble regroupe des éléments ayant des caractéristiques communes.

Le but des algorithmes de clustering est donc de minimiser la distance intra-classe (grappes d'éléments homogènes) et de maximiser la distance inter-classe afin d'obtenir des sous ensembles le plus distincts possible.[14]

La mesure des distances est un élément prépondérant pour la qualité de l'algorithme de clustering. Parmi ces méthodes la méthode des

6.2.1.1. L'algorithme de k-moyennes (k means):

Il permet de partitionner une collection d'objets en K groupes homogènes appelés clusters. Le nombre de clusters K est fixé par l'utilisateur selon ses attentes. L'algorithme des k-moyennes se déroule de la façon suivante : [7]

1. Choisir k objets formant ainsi k clusters
2. (Ré) affecter chaque objet O au cluster C_i de centre M_i tel que dist (O, M_i) est minimal
3. Recalculer M_i de chaque cluster (le barycentre)
4. Aller à l'étape 2 jusqu'à aucun item trouver.

6.2.2. Les règles associatives :

Parmi les travaux de recherche en fouille de données, l'extraction des règles d'association est la tâche phare qui a attiré plus d'attention des chercheurs et 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 [4].

Les règles associatives sont des règles qui sont extraites d'une base de données transactionnelles (itemset) et qui décrivent des associations entre certains éléments. Cette technique permet de faire ressortir les associations entre les produits de base (les produits essentiels, ceux pour lesquels le client se déplace) et les produits complémentaires, ce qui permet de mettre en place des stratégies commerciales visant à accroître.[14]

Des exemples de règles d'association:

- Si un client achète des plantes alors il achète du terreau,
- Si un client achète une télévision, il achètera un magnétoscope dans un an.

6.2.3. Les motifs séquentiels (en anglais sequence mining):

Le sequence mining concerne la détection de motifs dans les flux de données dont les valeurs sont délivrées par séquences.

Cette technique est particulièrement utilisée en biologie pour l'analyse de gènes et des protéines mais également afin de faire du text mining, les phrases étant considérées comme des séquences ordonnées de mots.[14]

7. Domaines d'application :

Dans nos jours le fouille de donnée est devenu plus en plus applicable dans différentes domaines et secteurs d'activités parmi lesquels :

Domaine des assurances :

- analyse des risques (caractérisation des clients à hauts risques, etc.)
- automatisation du traitement des demandes (diagnostic des dégâts et détermination automatique du montant des indemnités)
- consentements de prêts automatisés, support à la décision de crédit
- consentements de prêts automatisés, support à la décision de crédit détection des fraudes

Grande distribution :

- profils de consommateurs et modèles d'achats
- constitution des rayonnages marketing ciblé

Gestion et analyse des marchés :

- Analyse des comportements des consommateurs
- recherche de ses similarités en fonction de critères
- prédiction des ventes croisées
- optimisation des réapprovisionnements. [4]

Laboratoires pharmaceutiques :

- Modélisation comportementale et prédiction de médications ou de visites
- identification des thérapies pour différentes maladies.

Banques :

- recherche de formes d'utilisation de cartes caractéristiques d'une fraude
- prédictive des clients partants
- détermination de pré-autorisations de crédits.

Assurance :

- modèles de sélection et de tarification
- analyse des sinistres, recherche des critères explicatifs du risque ou de fraude
- prévision d'appels sur les plates formes d'assurance directe Aéronautique, Automobile et industries : contrôle de qualité des défauts
- prévisions des ventes, dépouillement d'enquête de satisfaction.

Télécommunications, eau et énergie :

- Simulation des tarifs
- détection des formes de consommation frauduleuses
- classification des clients selon la forme de l'utilisation des services prévisions de ventes.

Education :

- Analyse des facteurs d'échecs…

8. Conclusion :

La fouille de données est un domaine de recherche en plein essor visant à exploiter les grandes quantités de données collectées chaque jour dans divers domaines d'application de l'informatique. Après cette présentation générale de la fouille de données , nous avons détaillé les principales techniques utilisées dans ce dernier, et nous avons présenté le domaine de Data Mining, le chapitre suivant sera sur la technique des règles d'association.

CHAPITRE 02

LES REGLES D'ASSOCIATION

1. Introduction :

L'extraction des règles d'association est devenue aujourd'hui l'une des tâches les plus populaires de la fouille de données, ce concept a été introduit en 1993 par Agrawal [20]. les règles d'association représentent un outil efficace et performant qui a fait ses débuts dans le domaine de l'analyse du panier de la ménagère, cette tâche 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 [21].

Dans ce chapitre, nous avons essayé de présenter la technique de la fouille de données qui elle est les règles d'association avec leur domaine d'application, ses algorithmes et leurs mesures de qualité.

2. Les règles d'association :

Les règles d'association sont une tâche d'extraction des informations à partir de la coïncidence inter les données, il s'agit d'obtenir des relations ou des corrélations entre des ensembles d'items dans un grand volume des bases de transaction. Autrement dit, étant donnée un ensemble d'items, le but est de découvrir si l'occurrence de cet ensemble dans une transaction est associée à l'occurrence d'un autre ensemble d'items de la forme « **SI** A (condition ou antécédent) existe, **ALORS** il est possible que B (résultat ou conséquent) le soit aussi ». Par exemple *: «80% des clients qui achètent un ordinateur achètent aussi une imprimante et un abonnement à Internet» est une règle d'association associant l'item ordinateur aux items imprimante et abonnement à Internet ».* [22]

L'extraction des règles d'association se base principalement sur deux mesures, le support et la confiance. En effet, la plupart des algorithmes utilisés dans ce domaine, parcourent les données pour trouver les éléments qui dépassent un support minimum défini par l'utilisateur, et extraire par la suite, les règles d'association dont la confiance dépasse une confiance minimum. [23]

Les règles d'association produites par la méthode peuvent être facilement utilisées dans le système d'information de l'entreprise. Cependant, il faut noter que la méthode, si elle peut produire des règles intéressantes, peut aussi produire des règles triviales (déjà bien connues des intervenants du domaine) ou inutiles (provenant de particularités de l'ensemble d'apprentissage) [11].

2.1. Domain d'application les règles d'association :

Plusieurs systèmes de KDD utilisant l'extraction de règles d'association ont été utilisés pour des applications réelles dans divers domaines tels que le marketing, l'aide au diagnostic médical, les télécommunications, la téléphonie, etc, afin d'améliorer leurs résultats dans leurs activités. Son large champ d'applications rend la recherche d'associations un sujet de recherche attractif et très actif.

Nous présentons une listes non exhaustive des applications dont les résultats ont put être améliorées par l'analyse des règles d'association extraites.[24]

- **Planification commerciale :**

  L'identification des articles achetés fréquemment ensemble apporte une aide importante dans le placement des articles.

  La définition de catalogues. Les règles d'association permettant aux sociétés de vente par correspondance de déterminer quels articles il est préférable de placer sur la même page d'un catalogue.

  Déterminer quels articles en promotion pourront inciter les clients à effectuer d'autres achats.

  Dans le cas de transactions, les règles d'association permettant de définir des catalogues personnalisés en se basant sur les achats précédents du client [24].

- **Recherche médicale :**

  La plupart des organismes médicaux (hôpitaux, laboratoires d'analyse, cabinets médicaux, etc.) stockent systématiquement les informations relatives à leurs patients dans des bases de données. Ces informations sont les résultats de consultations auprès des médecins, les résultats de mesures indiquant la conditions des patients et des données sur l'évolution de la condition du patient pendant le traitement. L'extraction de règles d'association dans ces bases de données permet :

  D'apporter une aide au diagnostic en identifiant les symptômes ou maladies précurseurs d'une maladie.

  Déterminant les symptômes ultérieurs ou les effets secondaires possibles.

  L'identification de populations à risque vis-à-vis de certaines maladies.

D'identifier les analyses fréquemment pratiquées sur les mêmes patients, et de prédire les résultats de certaines analyses par combinaison de caractéristiques des patients et de résultats d'autres analyses[24].

- **Multimédia et internet :**

L'extraction des règles d'association a partir de données multimédia a donné lieu à de nombreuses études. Principalement dans le cadre de l'analyse d'images. Les applications concernent la reconnaissance militaire, le filtrage des données parasites, la prévision météorologique, l'imagerie médicale, l'aide dans les enquêtes criminelles, etc.

De même, un grand nombre de ressources sont accessibles par le réseau internet et un nombre important d'accès à ces données sont réalisés chaque jour par des millions d'utilisateurs. La taille et le nombre croissants des sites internet entrainent d'importants besoins d'outils pour :

La réorganisation de ces sites en fonction des cheminements des usagers.

L'aide à la navigation dans les systèmes de gestion d'informations.

La recherche et la sélection des sites (moteur de recherche). À partir des historiques des accès par les usagers aux ressources des sites Internet ont été utilisées dans ce cadre pour l'aide à la conception et l'organisation des sites [24].

- **Analyse de Données statistiques :**

Constitue un défi important pour le KDD de par sa difficulté. La difficulté provient de la nature des données statistiques, qui sont fortement corrélées et dense, ce qui pose d'important problème d'efficacité. L'intérêt tient au nombre d'applications pouvant bénéficier de l'analyse des données statistiques qu'elles utilisent. Les organismes financiers, de recherche et les administrations stockent de nombreuses données de ce type (résultats de recensements, de sondages et d'études par exemple). les règles d'association peuvent constituer des indicateurs utiles dans ce cadre [24].

2.2. Les étapes d'extraction des règles d'association:

L'extraction des règles d'association peut être décomposée en quatre étapes qu'illustre la Figure 2.1 :[25]

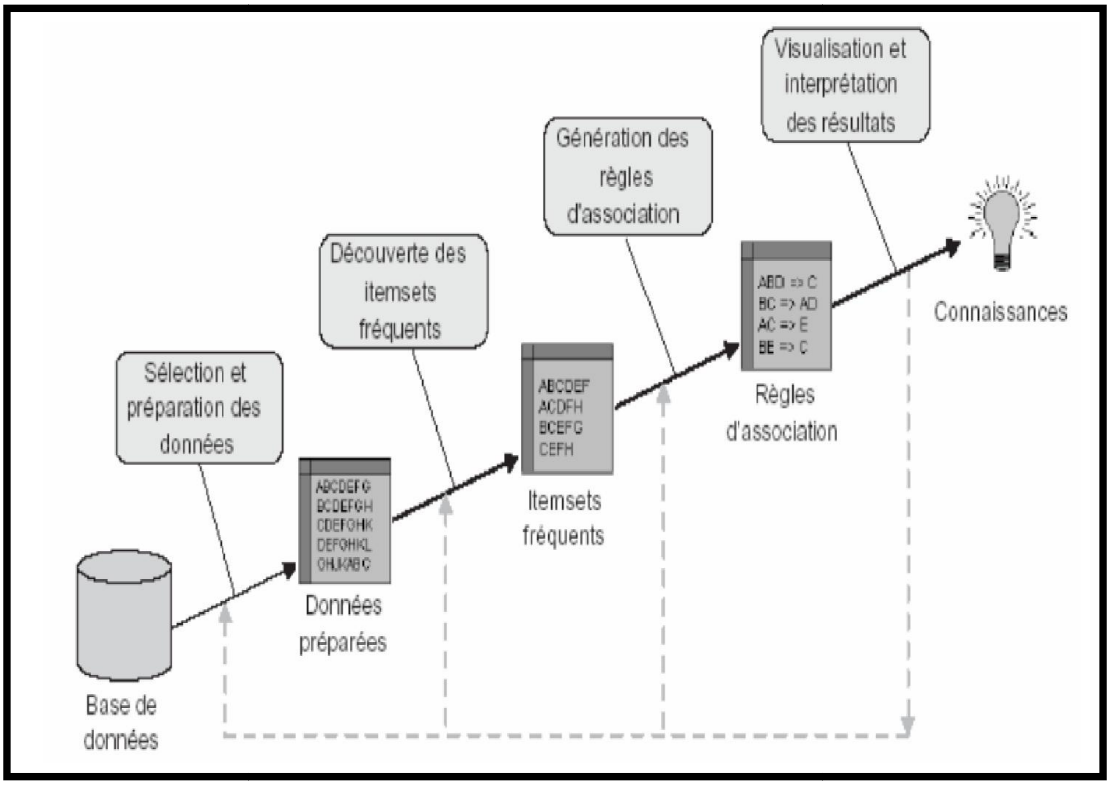


Figure 2.1. Les étapes d'extraction les règles d'association (abdelali mouad, 2003)

2.2.1. Sélection et préparation des données :

Cette étape permet de préparer les données. Elle est constituée de deux phases :

- La sélection des données de la base qui permettront d'extraire les informations intéressant l'utilisateur. Ainsi la taille des données traitées est réduite ce qui assure une meilleure efficacité de l'extraction.
- La transformation de ces données en un contexte d'extraction (il s'agit d'un triplet constitué d'un ensemble d'objets, d'un ensemble d'itemsets et d'une relation binaire entre les deux). La transformation des données sélectionnées en données binaires améliore l'efficacité de l'extraction et la pertinence des règles d'association extraites.[24]

2.2.2. Découverte des itemsets fréquents :

Un itemset fréquent est un ensemble d'éléments dont le support est supérieur ou égal à un certain support minimal spécifié par l'utilisateur. Cette étape est très coûteuse en temps d'exécution. Pour un ensemble de n items par exemple, le nombre d'itemsets fréquents qui peut être générés est de 2^n.[23].

2.2.3. Génération des règles d'association :

La génération des règles d'association consiste à déterminer les règles d'association dont le support et la confiance sont supérieurs ou égaux à un certain support et confiance minimaux définis par l'utilisateur.[23]

2.2.4. Visualisation et interprétation des règles d'associations :

Elle met entre les mains de l'utilisateur un ensemble de déductions fiables qui peuvent l'aider à prendre une décision [24].

3. Concepts généraux :

Dans la section qui suit, nous présentons les algorithmes d'extraction les règles d'association, mais bien avant et pour la clarté nous donnons quelques définitions pour mieux comprendre cette tâche.

3.1. Définition :

Cette section est consacrée à la définition de plusieurs termes utilisés dans la recherche et l'extraction des règles d'association :

- **Item:**

  Est tout objet, article, attribut, littéral, appartenant à un ensemble fini d'éléments distincts $I = \{x_1, x_2, ..., x_m\}$. Dans les applications de type analyse du panier de la ménagère, les articles en vente dans un magasin sont des items. [26]

- **Itemset:**

  Est un ensemble de n Items. L'ensemble de tous les Itemsets possiblement formés par les éléments d'Items est 2^n.[27].

- **Transaction :**

  Est un ensemble d'items par exemple les items achetés par un client C à une date précise. Dans une base de données une transaction est représentée par trois attributs : idClient (identifiant d'un client), idDate (un identifiant pour une date), itemset (un ensemble d'items non vide).[25]

- **Basse de donnée transactionnelle :**

  Une base de données transactionnel le peut être représentée sous forme horizontale, verticale ou binaire.[26]

Exemple :

TID	Lait	Banane	Café	Pizza	Sucre
1	1	0	0	1	0
2	1	1	1	0	1
3	1	0	1	0	0
4	0	1	0	1	0
5	1	1	1	1	0

Table 2.1. Représentation binaire des données de paniers de clients [3]

- **Support d'un Itemset :**

Le support d'un Itemset $\boldsymbol{I_K}$ noté **supp** (I_K, T) est la probabilité de qu'une transaction t_i contienne cet Itemset I_K. Le nombre de transactions incluant I_k divisé par le nombre total des transactions [23].

Exemple :

$$support(Lait) = P(Lait) = \frac{4}{5} = 80\%$$

Le support d'un k-itemset est supérieur ou égal au support d'un (k+1)-itemset le contenant. Ce (k + 1)-itemset est appelé *superset* du k-itemset [28].

✓ En fonction de leur support, les itemsets sont dotés de caractéristiques :

- Un itemset est dit clos ou fermé si aucun de ses *supersets* n'a de support identique au sien. Cela signifie que tous ses *supersets* ont un support inférieur ;
- Un itemset est dit maximal si aucun de ses *supersets* n'est fréquent ;
- Un itemset est dit générateur si tous ses sous-sets ont un support supérieur [28].

- **Itemset fréquent :**

Un Itemset I_K est fréquent si et seulement si son support est supérieur à un support minimum défini par l'utilisateur [23].

✓ A partir de ces définitions, Bayardo & Agrawal [28] ont introduit la notion de bordure (Border) pour caractériser la séparation entre les itemsets fréquents et les itemsets non fréquents :

• *Une bordure positive* est l'ensemble des itemsets fréquents maximaux, c'est-à-dire qu'ils n'ont pas de supersets fréquents ;

• *Une bordure négative* est l'ensemble des itemsets non fréquents, dont les sous-sets de premier ordre inférieur sont des itemsets fréquents.

La bordure sépare donc l'ensemble des itemsets en deux sous-ensembles, celui des itemsets fréquents, et celui des itemsets non fréquents. Elle est exploitée par des algorithmes de recherche de règles d'association. [28]

Treillis :

Un ensemble ordonné (T_r, <=) est un treillis si toute paire d'éléments de T_r possède une borne inférieure et une borne supérieure.[3] Exemple :

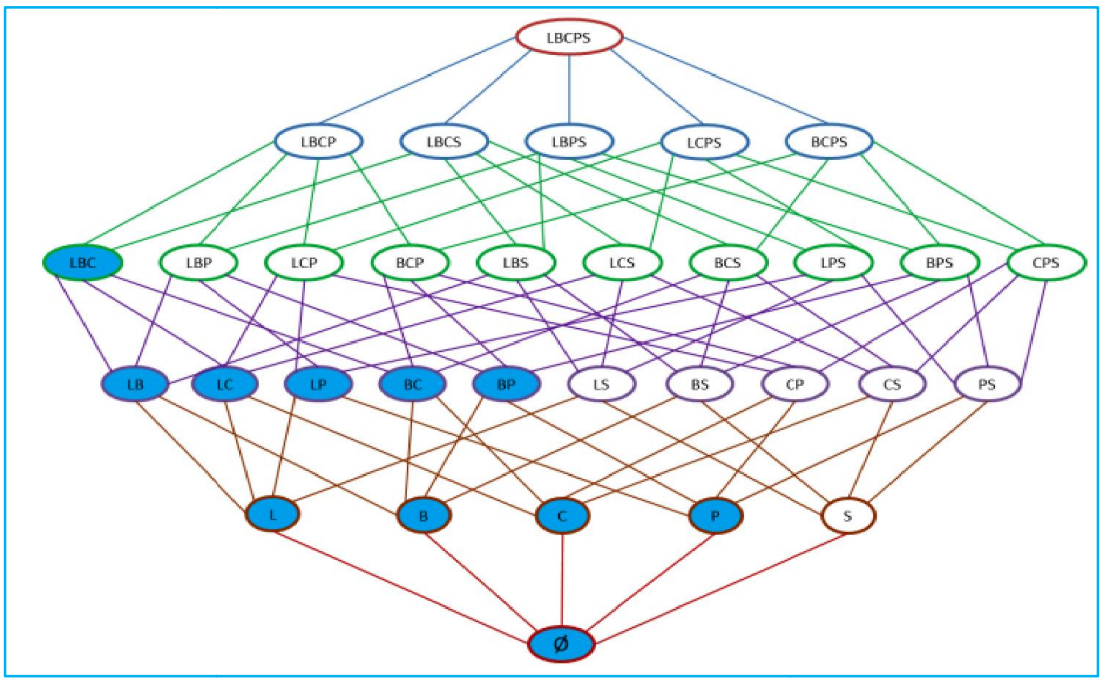


Figure 2.2. Exemple de treillis associé à un ensemble de 5 items [3]

Un exemple de treillis est illustré dans la figure 2.2. Ce treillis des motifs représente les données du "panier" (décrites dans la table 2.1). C'est en effet une représentation visuelle d'un ordre fini de l'ensemble des motifs selon la relation d'inclusion ensembliste. Nous désignons par les lettres L, B, C, P et S, respectivement les articles Lait, Banane, Café, Pizza et Sucre.

- **Support minimal :**

Notée minsup(I_i) est le nombre minimum d'occurrence d'un Itemset pour être considéré comme fréquent. L'occurrence n'est prise en compte q 'une fois dans la transaction. C'est un seuil choisi par l'utilisateur. [25]

- **Une règle d'association:**

Une règle associative ou règle d'association R, est définie comme une implication de la forme R: $I_1 \rightarrow I_2$, tel que $I_1 \subset I$, $I_2 \subset I$ et $I_1 \cap I_2 = \emptyset$, I_1est appelé "Condition" et I_2"Conclusion", elle déterminer par un support et une confiance.[4]

- **Le support d'une règle :**

notée Support(R), correspond au rapport du nombre de transactions, où apparaissent simultanément les items de la condition et du conclusion, sur le nombre de transactions total. Le support permet de mesurer la fréquence de l'association. [20]

$$support(X \rightarrow Y) = P(XY) = \frac{|T_{XY}|}{|T|}$$

- **La confiance d'une règle :**

notée Confiance(R), correspond au rapport du nombre de transactions où apparaissent simultanément les items de la condition et de la conclusion sur le nombre de transactions où apparaissent simultanément les items de la condition.

La confiance permet de mesurer la force de l'association. [20]

$$confiance(X \rightarrow Y) = P(Y|X) = \frac{P(XY)}{P(X)} = \frac{|T_{XY}|}{|T_X|}$$

La règle $X \rightarrow Y$ est dite "pertinente" ou "satisfaite", dans l'ensemble des transactions T, si sa confiance est supérieure à une confiance minimale Minconf fixée par l'utilisateur.[20]

4. L'extraction les règles d'association :

Les techniques d'extraction les règles d'association se déroulent en deux phases :[27]

1) extraction des Itemsets fréquents.
2) génération des règles d'association confiantes.

4.1. extraction des Itemsets fréquents :

Cette phase constitue la première partie de problème de recherche de règles d'association, il existe trois grandes approches algorithmiques pour la recherche d'itemsets fréquents pour la génération des règles d'association. [29]

- Approche d'extraction d'itemsets fréquents.
- Approche d'extraction d'itemsets maximaux.
- Approche d'extraction d'itemsets fermés fréquents.

4.1.1. Approche d'extraction des itemsets fréquents :

Dans (Agrawal, 1994) les ensembles extraits a l'étape 1 étaient appelés ' large itemsets', nom qui à été changé en 'fréquent itemsets ' dans leur papier suivant (Agrawal, 1996). C'est ce dernier, adopté par la communauté des Data-Miners.[30] Consiste à dégager les itemsets dont le nombre d'occurrences est supérieur au seuil minimum de support.

L'étape d'extraction des itemsets fréquents est la phase la plus coûteuse en temps d'exécution. Elle présente une complexité exponentielle en fonction de la taille de la base des transactions. Pour un ensemble de n items par exemple, le nombre d'itemsets fréquents qui peut être générés est de 2^n [30].

Les algorithmes d'extraction des itemsets fréquents sont basés sur les intuitions, démontrées dans (Pasquier, 2000), suivantes :

- Tous les sous-ensembles d'un itemset fréquent sont eux-mêmes fréquents.
- Tous les sur-ensembles d'un itemset non fréquent sont non fréquents.

4.1.2. Approche d'extraction d'itemsets maximaux :

Y a-t-il un moyen de trouver rapidement ce i-itemset non fréquent sans considérer ces sous-ensembles?

Cette idée est à la base des fréquents maximaux (Kedem, 1998) qui sont caractérisés par leur recherche dans les transactions en combinant, une recherche ascendante, pour les itemsets de plus en plus grands, et une recherche descendante, pour les itemsets de plus en plus petits.

Le principe de leur recherche est le suivant :

Si, par le biais de la recherche descendante, un itemset a été classé fréquent, il est inutile de poursuivre la recherche vers ses sous-ensembles dont on est sûr qu'ils sont fréquent.

D'un autre côté, si par le biais de la recherche ascendante, un itemset a été classé non fréquent, il est inutile de poursuivre la recherche vers ses sur-ensembles dont on est sûr qu'ils sont non fréquents [30].

4.1.3. Approche d'extraction d'itemsets fermés fréquents :

Toujours dans leur tentative de réduire l'espace de recherche des fréquents, certains travaux (Pasquier, 1999) (Zaki, 2002) (Agrawal, 1996) ont introduit la notion de fermé (donc de fermé fréquent).

Le fondement théorique de la caractérisation des fermés est le suivant. Si l'on met en correspondance, grâce à une fonction f, l'ensemble des items avec l'ensemble des transactions où ils apparaissent. Puis on met en correspondance, grâce à une fonction h, l'ensemble des transactions avec l'ensemble des items qu'elles contiennent, (Pasquier, 1999).

L'ensemble des fermés, munis de la relation d'inclusion forment un treillis de Galois.

Un résultat important de (Pasquier, 1999) dit que l'ensemble des fréquents maximaux est identique à l'ensemble des fréquents maximaux fermés [30].

4.2. génération des règles d'association :

La génération des règles est donc une opération de transformation des ensembles d'items en règles de manière efficace , elle consiste à générer des règles d'association à partir de ces itemsets fréquents qui respectent un seuil minimum de confiance. Cette seconde étape est beaucoup moins coûteuse que la génération des itemsets fréquents, car il n'est plus nécessaire de faire des parcours coûteux de la base de transactions. Néanmoins, cette phase reste tout de même exponentielle dans la taille des itemsets fréquents, car le nombre de règles pouvant être générées à partir d'un k-itemset de taille supérieure à 1 est égal à $2^k - 1$ [27].

5. Algorithme d'extraction des règles d'association :

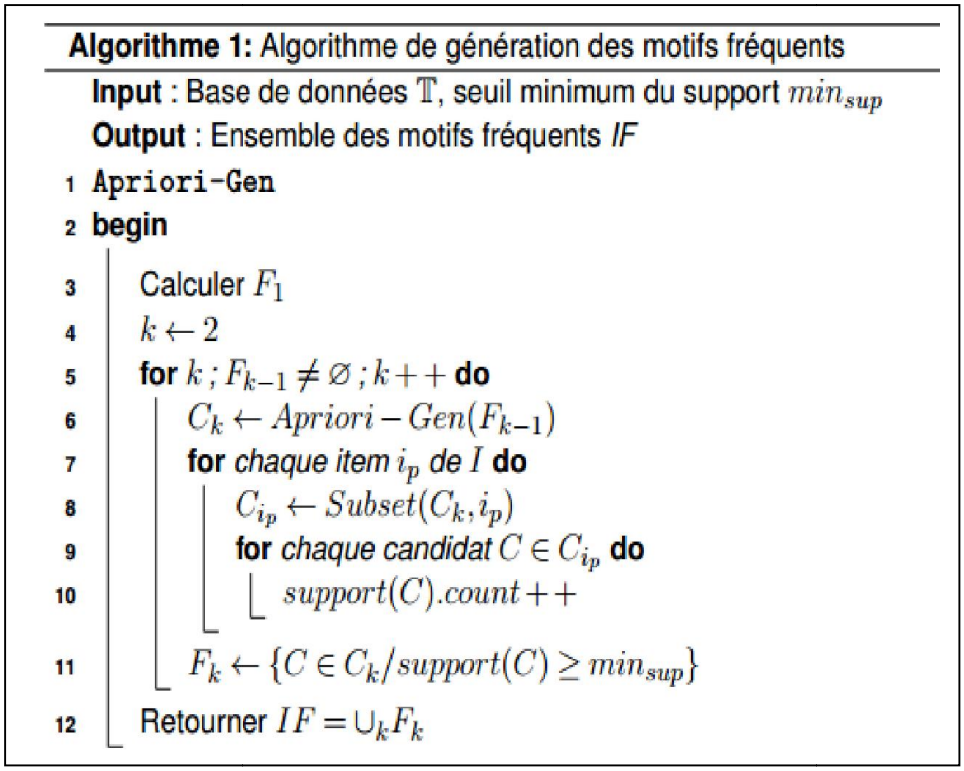
Dans cette section, nous essayons de présenter quelques algorithmes d'extraction de règles association les plus utilisés.

5.1. Algorithme Apriori :

Auteure : Agrawal et Srikant

Date de publication : 1994

Description : Cet algorithme fut une évolution dans l'histoire de l'extraction des règles d'association, Comme tous les algorithmes de découvertes d'associations, il travaille sur des bases de données transactionnelles (des enregistrements de transactions) [3].

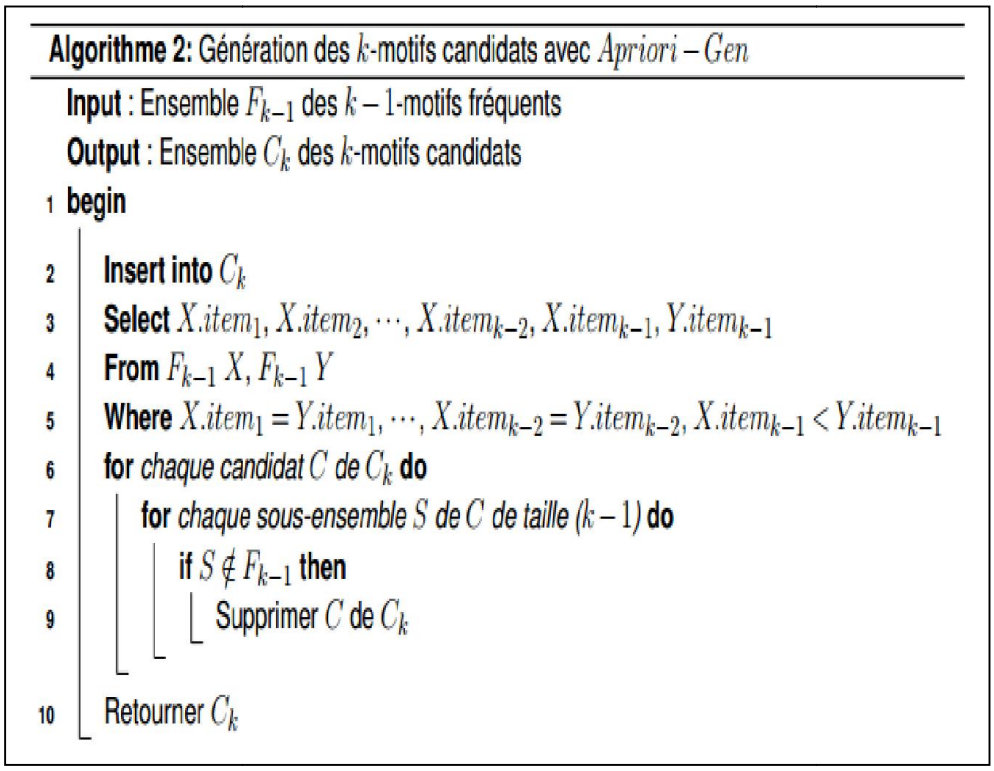


Algorithme 1: Algorithme de génération des motifs fréquents

Input : Base de données $\mathbb{T}$, seuil minimum du support min_{sup}
Output : Ensemble des motifs fréquents *IF*

```
1  Apriori-Gen
2  begin
3      Calculer F_1
4      k ← 2
5      for k ; F_{k-1} ≠ ∅ ; k++ do
6          C_k ← Apriori − Gen(F_{k-1})
7          for chaque item i_p de I do
8              C_{i_p} ← Subset(C_k, i_p)
9              for chaque candidat C ∈ C_{i_p} do
10                 support(C).count++
11         F_k ← {C ∈ C_k / support(C) ≥ min_sup}
12     Retourner IF = ∪_k F_k
```

Figure 2.3. Pseudo code de l'algorithme Apriori.



Algorithme 2: Génération des k-motifs candidats avec $Apriori-Gen$

Input : Ensemble F_{k-1} des $k-1$-motifs fréquents
Output : Ensemble C_k des k-motifs candidats

```
1  begin
2      Insert into C_k
3      Select X.item_1, X.item_2, ···, X.item_{k-2}, X.item_{k-1}, Y.item_{k-1}
4      From F_{k-1} X, F_{k-1} Y
5      Where X.item_1 = Y.item_1, ···, X.item_{k-2} = Y.item_{k-2}, X.item_{k-1} < Y.item_{k-1}
6      for chaque candidat C de C_k do
7          for chaque sous-ensemble S de C de taille (k−1) do
8              if S ∉ F_{k-1} then
9                  Supprimer C de C_k
10     Retourner C_k
```

Figure 2.4. Pseudo code de l'algorithme Apriori-Gen.

L'algorithme 1 ainsi introduit permet de découvrir les itemsets fréquents en partant de ceux de taille 1(ligne 3), on note cet ensemble F_1 Chaque itération k (lignes 4 à 12) se subdivise en deux étapes :

1. La première étape fait appel à la procédure *Apriori − Gen*. *Apriori-Gen* Cette procédure, décrite dans l'algorithme 2, est aussi constituée de deux phases :

   a) La première phase nommée **Joindre** (lignes 2 à 5 de l'algorithme 2) permet de déterminer l'ensemble C_K des k-itemset candidats, i.e., les k-itemset qui sont potentiellement fréquents à partir des (k − 1)-itemset fréquents de F_{k-1}

   b) La deuxième phase nommée **Effacer** (ligne 6 à 9 de l'algorithme 2) consiste à supprimer de C_k les éléments qui ne vérifient pas la propriété d'anti-monotonie des sous-ensembles fréquents. Deux motifs X et Y de F_{k-1} forment un motif C si, et seulement s'ils ont (k − 2) attributs (dans le prefixe) en commun, ce qui est exprimé en utilisant l'ordre lexicographique (ligne 5)

2. La deuxième étape (lignes7à11de l'algorithme 1) fait appel à la procédure Subset (C_k, i_p).

   Subset Cette procédure détermine les k-itemset fréquents parmi les k-motifs candidats. Il s'agit de trouver pour chaque itemset i_P de l'ensemble I (ligne 7), l'ensemble Cip des k motifs candidats qu'il possède (au moyen de la procédure Subset, ligne 8). Ainsi, pour trouver le support de chaque candidat, il s'agit de parcourir la base de données. Une fois l'ensemble Ci_P déterminé, le support du candidat sera incrémenté (ligne 10). Parmi les candidats, seuls ceux qui ont le support supérieur à minsup sont retenus (ligne 11). Le processus s'arrête lorsqu'aucun nouveau itemset candidat ne peut être généré, i.e., lorsque $F_{k-1} = \emptyset$.

Disposant des motifs fréquents, il nous faut maintenant découvrir les règles d'association. Générer l'ensemble R des règles d'association à partir de l'ensemble IF de motifs fréquents trouvés durant la phase précédente (trouver les motifs fréquents). Pour chaque motif fréquent X, nous considérons tous ses sous-ensembles (d'après la propriété d'antimonotonicité, sont tous fréquents) pour générer toutes les règles $Y \rightarrow (X \setminus Y)(Y \subset X)$. Afin de limiter l'extraction aux règles d'association valides, seules celles qui possèdent une confiance supérieure ou égale au seuil minimum minconf sont retenue.

Plusieurs algorithmes proposent d'améliorer la recherche des règles d'association. Ils répondent à des besoins de performance, ou de rapidité d'exécution, et prennent également en

compte, par exemple, la possibilité de pouvoir charger intégralement la base en mémoire ou non, ou la possibilité de traiter de très longs itemsets, comme c'est le cas dans le domaine de la biologie.[28]

5.2. Algorithme FP-growth :

Auteurs : han, al.

Date de publication : 2000.

Description: FP-Growth (Frequent-Pattern Growth), utilise une structure de données compacte appelé Frequent-Pattern tree et qui apporte une solution au problème de la fouille de motifs fréquents dans une grande base de données transactionnelle. En stockant l'ensemble des éléments fréquents de la base de transactions dans une structure compacte, on supprimer la nécessiter de devoir scanner de façon répétée la base de transactions. De plus, en triant les éléments dans la structure compacte, on accélère la recherche des motifs [20].

Le processus d'extraction de patterns se déroule en deux grandes étapes :

- la construction de l'arbre appelée FP-Tree (une représentation condensée de la base de données) ;
- la génération des patterns fréquents à partir de cette structure, l'arbre FP-Tree.

Un FP-tree est une structure compacte constituée d'un :

- Arbre : mise à part la racine nulle, chaque nœud de l'arbre contient trois informations : l'item que représente ce nœud, sa fréquence, ainsi que le nœud suivant dans l'arbre.
- Index : contient la liste des items fréquents. A chaque item est associé un pointeur indiquant le premier nœud de l'arbre contenant cet item [4].

5.3. Algorithme Eclat :

Auteurs : Zaki et al

Date de publication : 1997

Description : cet algorithme utilise le format vertical de la base de données, ou pour chaque itemset on dispose de son tidset, i.e. de l'ensemble de toutes les transactions contenant cet itemset. Le format vertical a l'avantage de rendre le calcul du support plus simple puisqu'il s'agit d'effectuer dans ce cas des intersections des tidsets. De plus ceci

réduit automatiquement la taille de la base de données puisque seules les transactions concernant un itemset sont utilisées pour l'intersection [26].

Eclat effectue une recherche des itemsets fréquents en profondeur d'abord et se base sur le concept de classes d'équivalence, on considère que deux itemsets sont dans la même classe d'équivalence s'ils possèdent un préfixe commun.

Par exemple, les itemsets ABC et ABD sont dans la classe d'équivalence AB, chaque classe peut être traitée séparément en mémoire, ce qui permet de décomposer le treillis en sous-treillis où chaque sous-treillis représente une classe d'équivalence [4].

L'avantage de cette approche, comme le soulignent ses auteurs, est qu'elle reste facilement parallélisablé, étant donné que l'on peut chercher les itemsets fréquents dans les différentes classes d'équivalence séparément [26].

5.4. L'algorithme partition :

Auteurs : Savasere et al

Description : réduit le nombre de parcours de la base de données à deux parcours seulement. Comme son nom l'indique, le principe de Partition est de partitionner la base de transactions D en p partitions D_1, ….D_p, telles que chaque partition peut être chargée en mémoire. Les itemsets fréquents locaux à chaque partition sont trouvés lors d'un premier parcours de D [26].

L'algorithme se base ensuite sur la propriété stipulant qu'un itemset fréquent dans toute la base de données doit être fréquent dans au moins une partition de la base de données. L'algorithme se décompose en deux étapes : [4]

– recherche des itemsets fréquents locaux sur chaque Di

– Dans la deuxième étape un balayage est fait ; calculer les supports des itemsets fréquents globaux

6. Mesures de qualité des règles d'association :

Afin d'extraire les règles d'association d'une base de données, il est nécessaire de fixer le support seuil, pour déterminer les itemsets fréquents, ainsi que la confiance seuil, pour trouver les règles valides. En fonction du besoin de l'utilisateur, la taille maximale de la règle peut être fixée. Cependant, le support et la confiance ne sont pas toujours suffisants pour trouver des règles pertinentes [28]. En effet, en fonction des valeurs seuil, les algorithmes peuvent générer un nombre de règles très important, qui peut, dans certaines situations de seuil trop

bas, dépasser le nombre de transactions initiales. De même, si le minimum est trop élevé, alors des règles intéressantes à faibles supports peuvent ne pas être détectées. De plus, ces deux mesures sont souvent insuffisantes pour prouver l'intérêt d'une règle, parce qu'elles ne prennent pas en compte P (Y) ni les contre-exemples P (XY). Par exemple, si c(X ⇒ Y) = P (Y), cela signifie que X et Y sont indépendants, parce que P (X)P (Y) = P (XY). Cette règle n'est donc d'aucun intérêt, même si le support et la confiance sont élevés.

Il est donc nécessaire de caractériser les règles d'association par des mesures supplémentaires dites de qualités ou d'intérêt.[28]

- **Lift :**

  La mesure la plus connue est le lift, défini par :

$$l(X \Rightarrow Y) = \frac{P(XY)}{P(X)P(Y)}$$

  Le lift représente le rapport à l'indépendance de la règle X → Y. Pour cette règle, l'indépendance entre les attributs X et Y est égale à P (X) × P (Y). L'indice brut d'association réellement observé est lui égal à P (XY). Le lift permet donc d'apprécier simplement, pour une règle X → Y, sa « distance » à l'indépendance.

  Par exemple, une règle X → Y ayant un lift égal à 2 indique que les individus ayant la propriété X ont deux fois plus de chances d'avoir la propriété Y que les individus en général [31].

  Cette mesure est symétrique et ne permet donc pas de distinguer les règles X→ Y et Y → X.

- **Corrélation linéaire de Pearson :**

  Elle est définie par :

$$r(X,Y) = \frac{P(XY) - P(X)P(Y)}{\sqrt{P(X)P(\overline{X})P(Y)P(\overline{Y})}}$$

  Elle permet de mesurer la force de la liaison entre X et Y. Si elle est nulle, alors cela signifie que X et Y sont indépendants. Une valeur positive forte indique que X et Y sont corrélés. Une valeur négative forte indique que X et Y sont corrélés négativement, c'est-à-dire que X et not Y sont corrélés.

- **Loevinger :**

Elle est définie par :

$$LO(X \Rightarrow Y) = 1 - \frac{P(X\overline{Y})}{P(X)P(\overline{Y})} = \frac{P(Y/X) - P(Y)}{P(\overline{Y})}$$

L'une des plus anciennes mesures de qualité répertoriées dans le domaine de la fouille de données [31]. Cette mesure est considérée comme un indice d'écart à l'indépendance et prend la valeur nulle en cas d'indépendance. Elle augmente au fur et à mesure que le nombre de contre exemples diminue, c'est-à-dire quand P (XY) décroît, pour atteindre la valeur 1 quand il n'y en a plus. Elle décroît avec le support, et permet de rejeter des règles peu intéressantes, malgré une confiance élevée.

- **Confiance centrée :**

Elle est définie par :

$$CC(X \Rightarrow Y) = c(X \Rightarrow Y) - P(Y) = P(Y/X) - P(Y)$$

Dans le cas de l'indépendance, la confiance est égale à P (Y). En la recentrant par rapport à P (Y), la confiance centrée devient alors nulle à l'indépendance, ce qui est vrai quelle que soit la probabilité de Y.

Il y a pas mal des mesures permet d'évaluer la qualité des règles d'association, Cet état de l'art donne une idée de la diversité des techniques et mesures pour extraire des règles d'association, et pour les caractériser. Les valeurs seuil et les métriques sont autant de paramètres qu'il peut être intéressant de maîtriser et d'utiliser dans la recherche de solutions à un problème donné.

7. Classification des algorithmes :

Malgré que les algorithmes présentés utilisent différentes techniques et adoptent différents points de vue, ils peuvent être classés selon trois aspects algorithmiques: La stratégie de calcul du support, la direction de recherche, la stratégie de recherche [20].

7.1. Stratégie de Calcul du Support :

Le calcul du support des itemsets candidats peut se faire selon deux approches:

- **Calcul horizontal** (comptage direct d'occurrences): Détermine la valeur du support d'un itemset candidat en lui associant un compteur initialisé à 0 et en

parcourant la base de transactions, transaction par transaction. A chaque fois que l'on trouve l'itemset recherché dans la transaction courante, le compteur est incrémenté.

- **Intersection verticale:** une approche pour déterminer les supports des candidats est l'intersection verticale ou intersection des TidLists. Cette méthode est utilisée lorsque la base de transactions est représentée verticalement.

7.2. Direction de Recherche :

La direction de recherche réfère à la direction dans laquelle l'espace de recherche est traversé. La plupart des algorithmes d'extraction des itemsets fréquents, surtout ceux qui étendent Apriori, adoptent une traversée ascendante de l'espace de recherche, commençant par les 1-itemsets fréquents jusqu'à parvenir aux itemsets fréquents les plus larges (dans le sens de la taille)

L'autre type de traversée est la traversée descendante de l'espace de recherche, commençant par les itemsets fréquents les plus larges jusqu'à l'obtention des 1-itemsets fréquents. Cette stratégie est habituellement utilisée pour l'extraction des itemsets larges maximaux [20].

7.3. Stratégie de recherche :

Tandis que la direction de recherche guide la manière selon laquelle l'espace de recherche sera utilisé, la stratégie de recherche se réfère à l'ordre dans lequel sont visités les itemsets.

- La stratégie de recherche en largeur d'abord (BFS: Breadth First Strategy) opère niveau par niveau. Elle visite tous les itemsets de niveau (k-1) avant de visiter ceux de niveau k. La plupart des algorithmes qui étendent Apriori utilisent une recherche en largeur d'abord (BFS) parce qu'elle facilite l'élagage des candidats. Cette stratégie nécessite plus de mémoire pour garder les sous ensembles fréquents des candidats élagués.
- La stratégie de recherche en profondeur d'abord (DFS: Depth First Strategy), quant à elle, visite de façon récursive les descendants d'un itemset.

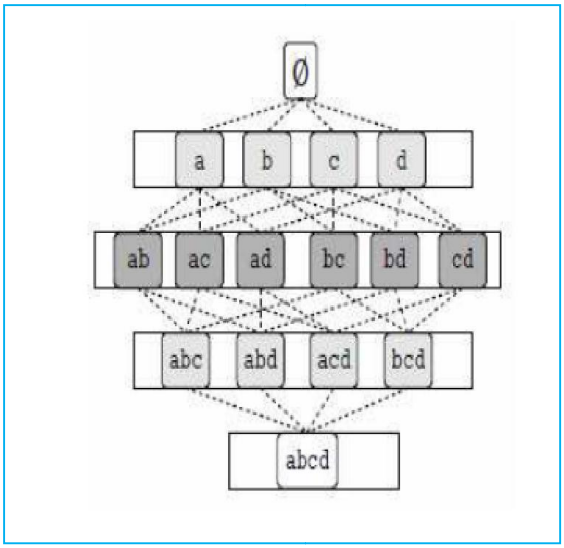


Figure 2.5. Recherche en largeur

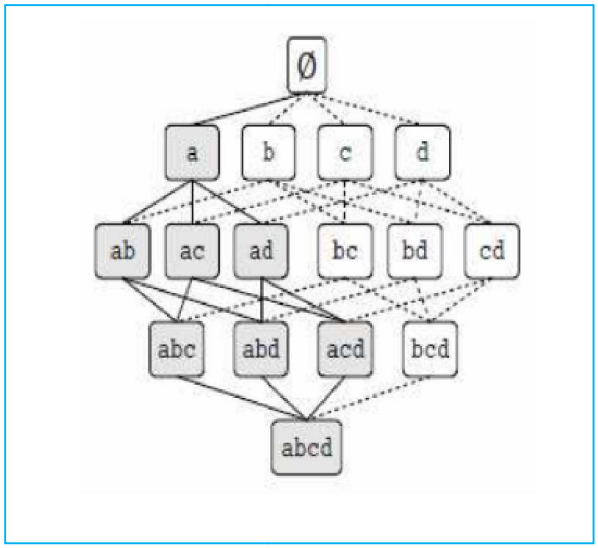


Figure2.6. Recherche en profondeur

La figure 2.7 montre la classification des algorithmes d'extraction des itemsets fréquents, précédemment présentés, selon les trois critères ci-dessus.

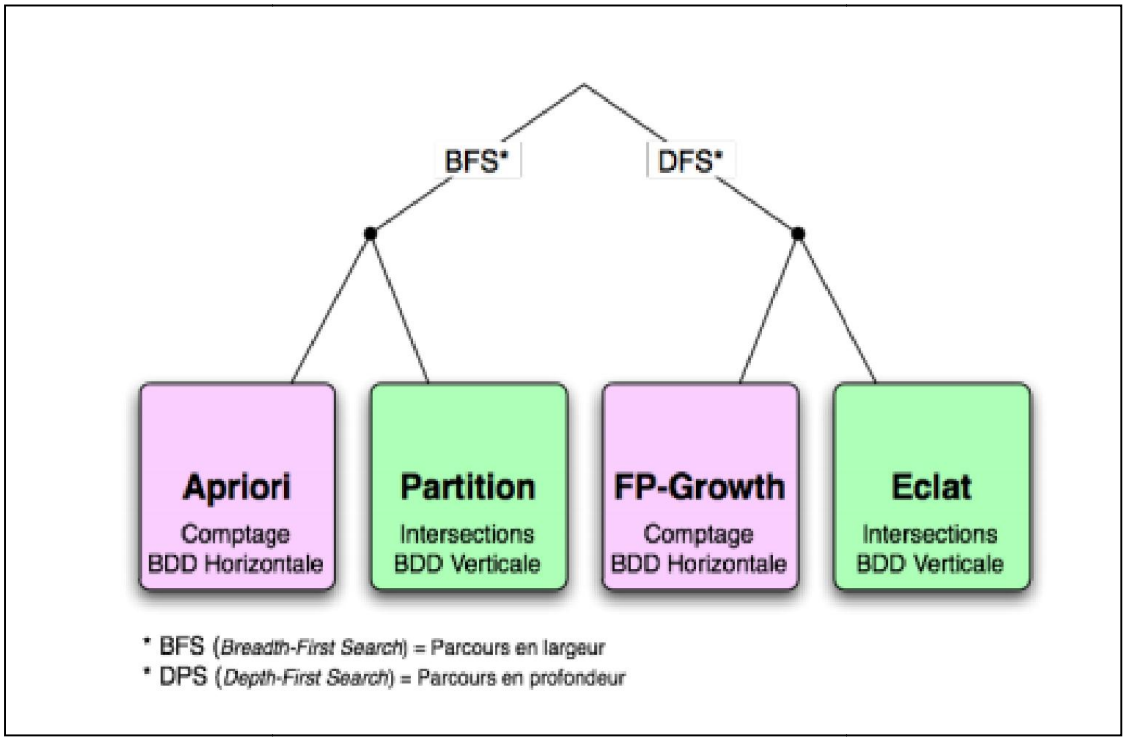


Figure 2.7. Classification des algorithmes d'extraction des règles d'association.

8. Conclusion :

Différentes techniques permettent de décrire les données, Même si toutes ces approches permettent d'extraire de la connaissance de grandes bases de données, elles ne sont pas (ou mal) adaptées `a l'extraction de comportements des données la problématique de la recherche de règles d'association est étendue pour détecter des comportements typiques dans le temps et le concept de motifs séquentiels est introduit

Dans la suite de ce travail, nous intéressons plus particulièrement a la découverte de motifs séquentiels.

CHAPITRE 03

LES MOTIFS SEQUENTIELS

1. Introduction :

Dans un certain nombre de domaines, la recherche de connaissances temporelles est non seulement utile mais nécessaire, Les approches classiques d'extraction de règles qui se basent sur la notion de motifs fréquents sont peu adaptées à ces problématiques, notamment celles qui s'intéressent à des associations fortes entre événements, sans que ces associations soient fréquentes. Les motifs séquentiels permettent de s'affranchir de la notion de fréquence et de sélectionner les règles en fonction de leur significativité statistique calculée à partir d'un modèle de durée. La recherche de telles informations devient d'autant plus intéressante qu'elle permet de prendre en compte un certain nombre de contraintes entre les évènements comme par exemple la durée minimale ou maximale séparant deux évènements.

Ce chapitre présente un état de l'art des approches et technique d'extraction des motifs séquentiels à partir de bases de données séquentielles, ces approches permettent d'extraire les comportements répandus qui ne sont pas visibles par des analystes humains.

2. motifs séquentiels :

Le problème de fouille de motifs séquentiels a été introduit pour la première fois par Agrawal et al. (1995) [7], La recherche de motifs séquentiels peut être vue comme une extension de la notion de règles d'association, intégrant des contraintes temporelles. Cette recherche met en évidence des associations entre les transactions (inter-transaction) par contre la recherche des règles d'association détermine les liens au sein d'une même transaction (intra-transaction). Les motifs séquentiels sont extraits à partir de séquences d'événements ordonnées et souvent sauvegardés dans des bases de données transactionnelles (voir l'exemple de la Table 3.1) [32]. Les motifs séquentiels ont pour objectif la découverte d'enchaînements fréquents dans les bases de données, avec des contraintes temporelles [2] et l'identification des événements d'individus afin de pouvoir suivre leurs comportements séquentiels au cours du temps [32].

2.1. Exemples d'applications possibles :

Plusieurs applications utilisent ce type de problème pour l'extraction des comportements séquentiels comme :

- Un système de fouille de Web peut découvrir des comportements comme :
  - *10% des personnes qui ont visité "/company/toys" ont fait dans un délai d'une semaine une recherche google d'après le mot "Disney".*

- *15% des personnes qui ont acheté le livre"/amazon/ Theorie_de_la_relativite" ont acheté dans les 10 jours d'après les livres "/amazon/Bibliographie_de_Einstein".*

▪ l'analyse du panier de la ménagère : qui étudie les comportements des clients en cherchant des séries d'articles qui sont fréquemment achetés dans un intervalle de temps donné.

- *36% des clients achètent une télévision, achètent un lecteur de DVD dans les deux ans qui suivent et un Home-Cinéma 6 mois après "*

▪ L'analyse de symptômes pour du diagnostique médical.

▪ l'analyse des réseaux de télécommunication.

▪ l'analyse des performances des systèmes.

▪ Analyse le comportement de profils d'internautes.

▪ La bioinformatique, etc .

2.2. Extraction de motifs séquentiels versus extraction d'itemsets et de règles d'association :

- La différence *théorique* entre les itemsets et les motifs séquentiels réside dans les contraintes temporelles. Alors que {pain , lait, figues} représente un itemset, {pain suivi par l'achat de lait et des figues dans les trois jours suivants} exprime un motif séquentiel.
- Mais il y a une différence *pratique* très importante qui complique l'adaptation d'un algorithme d'extraction d'itemsets à l'extraction de motifs séquentiels : si dans un itemset les items sont uniques, dans le cadre d'un motif séquentiel on peut avoir des répétitions d'items.
- En d'autres termes, l'extraction des règles d'association s'intéresse aux motifs intra-transactions, alors que l'extraction de motifs séquentiels s'intéresse à la recherche des motifs inter-transaction.
- La différence entre l'extraction de règles d'associations et l'extraction de motifs séquentiels consiste dans le fait que la deuxième s'applique juste sur une base de données dont les transactions ont été préalablement triées en ordre croissant en fonction du temps associé à chaque transaction.
- Dans sa forme la plus simple, l'extraction de motifs séquentiels n'impose pas de contraintes temporelles (intervalle de temps entre deux transactions), tant que les transactions sont faites par le même client.

- Alors que dans l'extraction de règles d'association, les transactions sont utilisées seulement pour compter le support, dans l'extraction des motifs séquentiels, un client peut contribuer à un motif candidat seulement quand on compte les supports des motifs candidats. Plus simplement, dans le premier cas on compte le pourcentage de transactions, alors que dans le deuxième on compte le pourcentage de clients **[2].**

3. Concepts généraux :

Cette section présente les définitions et les principes généraux dans l'extraction de motifs séquentiels.

3.1. Définitions :

- **Une séquence :** est une structure de données qui permet d'organiser un ensemble d'éléments grâce à une relation d'ordre entre ces éléments [33], C'est une suite de transactions chronologiquement ordonnées et se rapportant à un même sujet.

Une séquence utilise le principe de précédence c'est à dire chaque élément de la liste est précédé des éléments qui l'ont précédé dans les transactions d'un client donné [27].

Exemple : si un client achète des produits *a, b, c, d, e* et selon la séquence *S =<(a)(bc)(d)(e)>,* cela signifie qu'il a d'abord achète le produits *a*, puis les produits *b* et *c* ensemble, ensuite le produit *d* et finalement le produit *e*.

- **Base de données transactionnelles :** Une base de données de séquences D basée sur les items de I est un ensemble fini de paires (SID, T), appelées transactions, avec SID ∈ {1, 2, . . .} un identifiant et T ∈ T (I) une séquence construite sur I.

Client	Date	Items
C_1	01/04/2008	{Pain,Cola}
C_1	02/04/2008	{Chips,Pain}
C_1	04/04/2008	{Pain}
C_1	18/04/2008	{Pain,Yaourt}
C_2	11/04/2008	{Chips}
C_2	12/04/2008	{Chocolat}
C_2	29/04/2008	{Yaourt, Pain, Chocolat}
C_3	05/04/2008	{Chips,Pain}
C_3	12/04/2008	{Yaourt,Pain}
C_4	06/04/2008	{Chips}
C_4	07/04/2008	{Chips}
C_4	08/04/2008	{Yaourt}

Table 3.1. Exemple base de données de séquences.

- **Inclusion :** une séquence S'= <s'$_1$ s'$_2$…..s'$_n$> est une sous-séquence de S=<s$_1$ s$_2$ ….s$_n$> s'il existe des entiers a$_1$ <a$_2$<….<a$_j$<….<a$_m$ tels que s'$_1$ ⊆s$_{a1}$, s'$_2$ ⊆ s$_{a2}$,....., s'$_m$ ⊆ s$_{am}$. On dit que S'est incluse dans S.

Exemple : La séquence S'=<(a)(b, c)(d)> est incluse dans la séquence S=< (a, d, e)(g, h)(f)(b, c, e)(d, e, f) > car (a) ⊆ (a, d, e), (b, c) ⊆ (b, c, e) et (d) ⊆ (d, e, f). En revanche, <(a)(b)> 6 ⊆/< (a, b)> (et vice versa). Les deux séquences < (a)(b)> et < (a, b)> sont dites incomparables.

Remarque: La longueur d'une séquence S est fonction du nombre d'items contenu dans la séquence et non pas du nombre d'itemsets ou de transaction. [34]

Exemple : la séquence relative au client C_2 dans la table 3.1 est un 5-séquence.

- **Fréquence d'une séquence :** Une séquence est considérée fréquente, si le support de cette séquence est supérieur ou égal au support minimum. Celui-ci est introduit par le client afin de mesurer la pertinence d'une séquence [25].

- **Support d'une séquence :** est le pourcentage de clients qui supportent cette séquence.

3.2. Propriétés des séquences fréquentes :

Il existe trois propriétés principales qui sont des éléments déterminants pour l'extraction. Il faut noter que toutes ces propriétés sont déjà appliquées sur les règles d'association, l'anti-monotonie, monotonie et support des sous-séquences.

4. Extraction les motifs séquentiels :

L'identification des événements d'individus au cours du temps est non seulement utile mais nécessaire afin de pouvoir suivre leurs comportements séquentiels. Les méthodes existantes différentes essentiellement sur la manière de parcourir l'espace de recherche (largeur d'abord ou profondeur d'abord) et sur les structures de données utilisées pour indexer la base de données et faciliter une énumération rapide, afin de réduire le temps d'exécution ainsi que les besoins en espace disque ou en mémoire vive [35]. Les algorithmes d'extraction des motifs séquentiels peuvent être classés en trois grandes catégories :

1. Méthodes horizontales
2. Méthodes verticales
3. Méthodes par projection

Nous décrivons dans les sections suivantes ces trois catégories ainsi que les principaux algorithmes associés.

4.1. Méthodes horizontales :

4.1.1. La méthode GSP (Generalized Sequential Patterns) :

A été l'une des premières propositions pour résoudre la problématique des motifs séquentiels, elle proposé par (Srikant and Agrawal 1996) [32], elle fait appel aux principes de l'algorithme Apriori, utilisé pour l'extraction d'itemsets fréquents, elle repose sur un parcours par niveau de l'espace de recherche s'appuyant sur le paradigme générer- élaguer [36].

GSP est un algorithme basé sur la méthode générer-élaguer afin de minimiser le nombre de passes sur la base de données. La technique généralement utilisée par les algorithmes de recherche de séquences est basée sur une création de candidats (par I-extension « ajout dans le dernier itemset » et S –extension « ajout d'une sous séquence ») suivie du test de ces candidats pour confirmer leur fréquence dans la base [37].

Le GSP utilise deux paramètres : le min-gap et le max-gap. Le min-gap spécifie la borne inférieure de temps écoulé requis pour que les transactions d'une séquence soit considérées comme valides, alors que le max-gap spécifie la borne supérieure de temps écoulé. Si le temps écoulé entre deux itemsets d'une séquence est supérieur au max -

gap, celle-ci n'est pas considérée comme une séquence valide et par conséquent son support n'est pas augmenté.

L'extraction débute par un premier parcours de la base afin d'identifier l'ensemble des événements fréquents (les 1-séquences fréquent). Par la suite, à chaque itération (K>=1) deux étapes sont appliqués:[38]

1. La génération des candidats : tient compte de la propriété d'anti-monotonie du support, construit l'ensemble C_k des séquences candidates à partir des L_{k-1}.
   La génération de candidats C_K se fait par auto-jointure des fréquents L_{k-1} (fournir par l'itération précédente). Elle identifie à partir de L^2_{k-1} tous les couples des séquences (s, s') telles que s et s' soit équivalentes en enlevant à la première son premier événement et à la seconde son dernier événement. À partir de chaque couple une k-séquence est construite on ajoutant élément de s'à s la totalité des nouvelles séquences représente C_k (les k-séquences candidates).
2. La phase d'élagage, élimine à partir de C_k les séquences non fréquentes. Une première phase de sélection élimine les candidats qui contiennent des sous séquences non fréquentes éliminé à l'itération précédente. Par la suite, le calcul de support des séquences restantes est calculé, pour cela, la base est parcourue une foi.
   L'appel récursif de ces deux phases est stoppé lorsque l'une des deux conditions suivantes est vérifiée :
   1. il n'y a plus de fréquents (phase 2)
   2. aucun candidat n'est généré (phase 1)

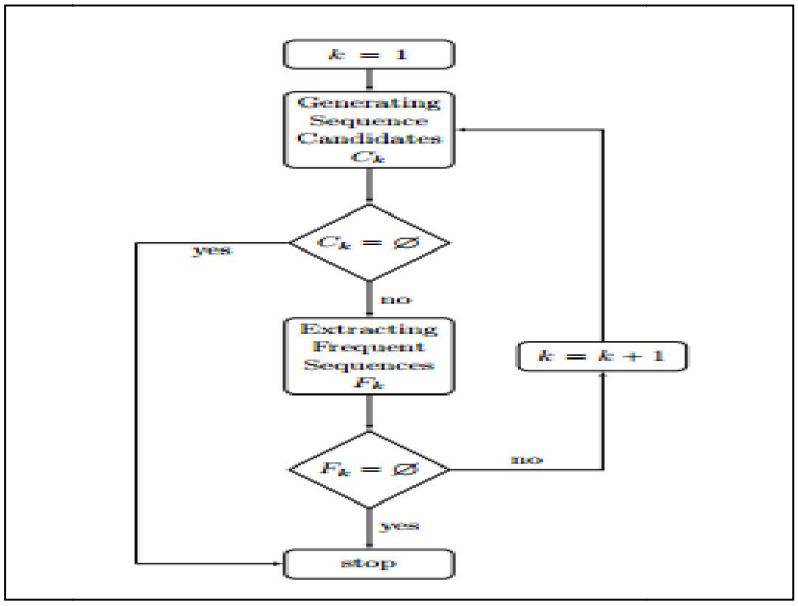


Figure 3.1 : diagramme représente l'algorithme GSP.

Pour évaluer le support de chaque candidat en fonction d'une séquence de données, GSP utilise une structure d'arbre de hachage (hash-tree) destinée à organiser les candidats. Les candidats sont stockés en fonction de leur préfixe. Pour ajouter un candidat dans l'arbre des séquences candidate, GSP parcourt ce candidat et effectue la descente correspondante dans l'arbre. Pour trouver quelles séquences candidates sont incluses dans une séquence de données, GSP parcourt l'arbre en appliquant une fonction de hachage sur chaque item de la séquence de données. Quand une feuille est atteinte, elle contient des candidats potentiels pour la séquence de données.

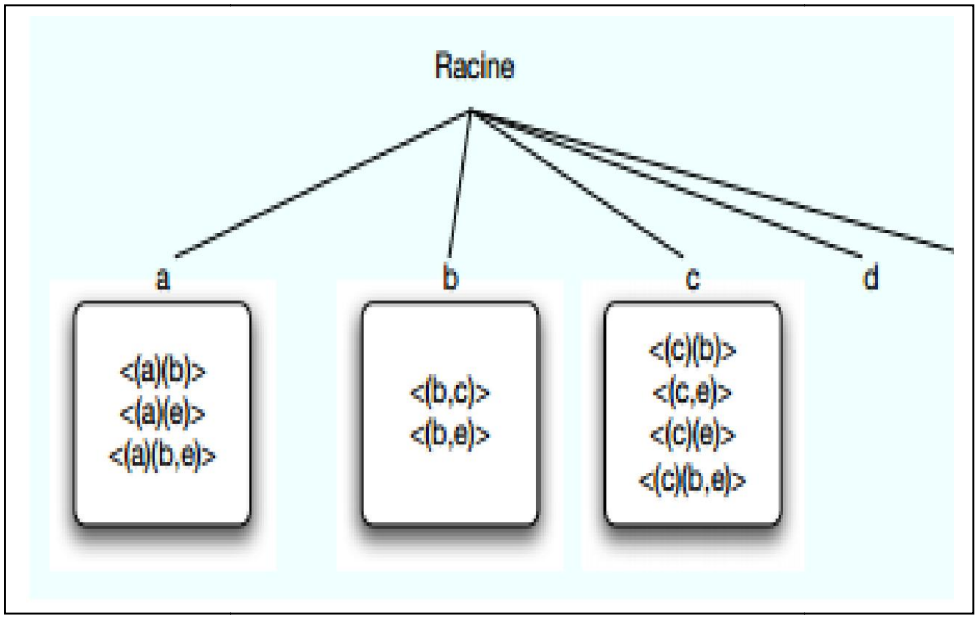


Figure 3.2 : La structure de données utilisée par l'algorithme GSP

4.1.1.1. *Limites de l'algorithme GSP* :

Le GSP possède toutefois des faiblesses :

1. Une grande quantité de candidats peut être générée dans les grandes bases de données. À titre indicatif, une base de données contenant 1000 1-séquences formera 1 499 500 candidats. Plusieurs de ces candidats ne se retrouveront pas dans la base de données, ce qui représente une perte de temps.
2. Une grande quantité de balayages de la base de données est requise. Étant donné que la longueur de chaque séquence candidate grandit d'un item à chaque balayage, l'identification d'une 15-séquence requiert 15 balayages de la base de données.

3. Les méthodes basées sur l'algorithme Apriori, comme c'est le cas pour le GSP, ont de la difficulté à découvrir de longues séquences. Ceci vient du fait que les longues séquences sont formées à partir d'un nombre important de séquences plus courtes et le nombre de candidats générés varie de manière exponentielle avec la longueur de ces derniers [25].

4.1.2. L'algorithme PSP (Prefix Tree for Sequential Pattern):

Il applique le même principe d'extraction que l'algorithme GSP. Toutefois, il en améliore les performances et met en place une structure hiérarchique différente pour représenter les candidats et permettre de prendre en compte le changement de temporalité entre les évènements [38], l'arbre de hachage utilisé par l'algorithme GSP présente un défaut. En effet lors de la recherche des feuilles susceptibles de contenir des candidats inclus dans la séquence analysée, la structure utilisée ne tient pas compte des changements de date entre les items de la séquence qui servent à la navigation.

Par exemple, avec la séquence < (10 30) (20 40)>, L'algorithme va atteindre la feuille du sommet 30 (fils de 10), alors que cette feuille peut contenir deux types de candidats :[39]

- ceux qui commencent par <(10) (30)... d'un côté.

- et ceux qui commencent par < (10 30) ... de l'autre

Le but est alors de mettre en place une structure d'arbre de préfixes, pour gérer les candidats.

Le principe de cette structure consiste à factoriser les séquences candidates en fonction de leur préfixe. De plus, pour prendre en compte le changement d'itemset, l'arbre est doté de deux types de branches. Le premier type, entre deux items, signifie que les items sont dans le même itemset alors que le second signifie qu'il y a un changement d'itemset entre ces deux items. Ainsi, l'arbre de préfixes ne stocke plus les candidats dans les feuilles mais permet de retrouver les candidats de la façon suivante : tout chemin de la racine à une feuille représente un candidat et tout candidat est représenté par un chemin de la racine à une feuille. Cette nouvelle structure de données permet ainsi une nette amélioration des performances lors de l'extraction de motifs séquentiels.[38]

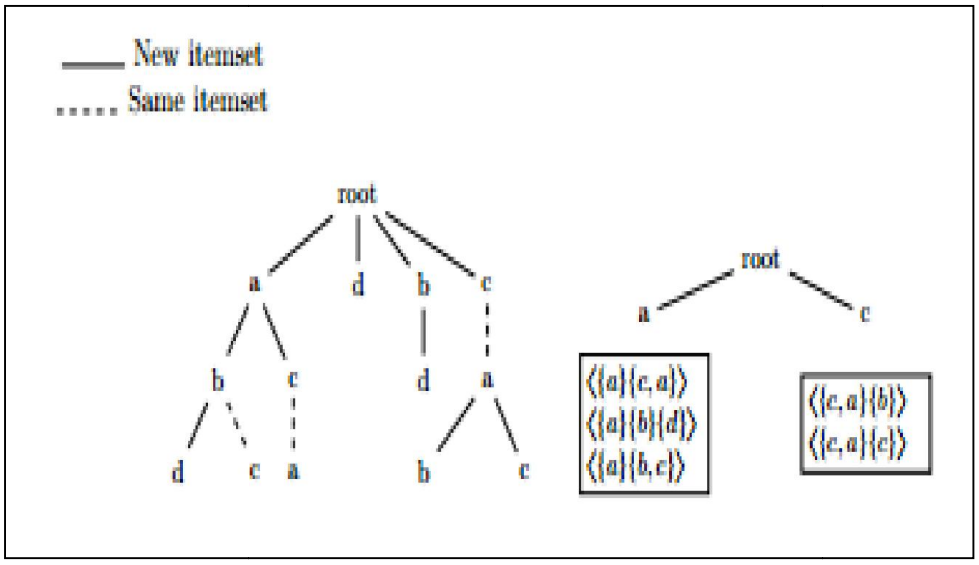


Figure 3.3 : l'arbre de préfix PSP (à gauche) et l'arbre de GSP (à droite)[35]

4.1.2.1. Limites de l'algorithme PSP:

Le principal problème de PSP est le nombre de passes dans la base de données. Pour une séquence de longueur k, k passes sont effectuées.

Ceci provoque une perte de temps lors de l'étape de comptage du nombre de clients qui supportent une séquence donnée. [25]

4.2. Méthode verticale :

A partir des années 1999 et 2000, les chercheurs dans le domaine de la fouille de motifs ont commencé à s'intéresser de plus en plus à l'optimisation de l'extraction. C'est dans ce contexte qu'est apparu l'algorithme SPADE.

4.2.1. L'algorithme SPADE :

SPADE (pour le terme anglais Sequential Pattern Discovery using Equivalence classes) ZAKI (2001), elle est le premier algorithme à proposer et utiliser une représentation verticale de la base de données pour extraire de manière plus efficace des motifs séquentiels [37]. Afin d'optimiser l'espace de mémoire utilisée, SPADE décompose l'espace de recherche dans des classes d'équivalences qui peuvent être traités indépendamment en mémoire [2], ces classes d'équivalences sont définies à partir d'une relation d'équivalence sur le préfixe. Ainsi, deux séquences de taille k sont dans la même classe d'équivalence si elles ont un préfixe commun de taille k – 1[37].

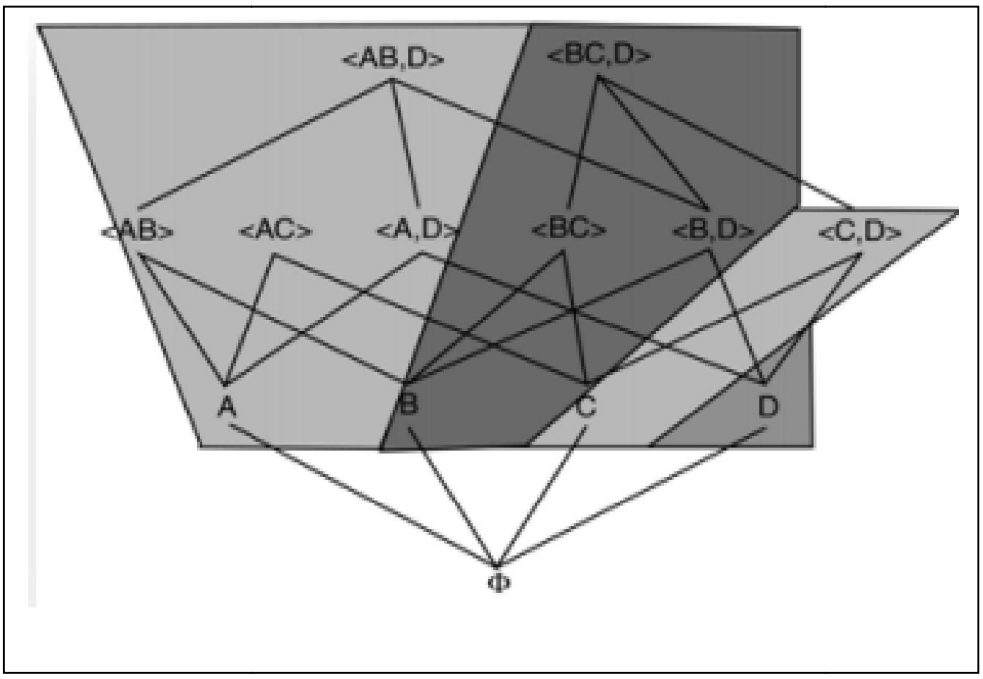


Figure 3.4 classes d'équivalences générer par l'algorithme SPADE [40].

Dans les bases de données verticales, la base de données devient un ensemble de nuplets de la forme <itemset : (sequence_ID, event_ID)>. L'ensemble des paires ID d'un itemset donné forme l'identifiant de la liste (ID_list) de l'itemset. Pour découvrir les k-séquences (séquences contenant k items), l'algorithme SPADE joint les ID_lists de deux éléments de l'ensemble des k − 1-séquences fréquentes.

SPADE a besoin de seulement deux balayages de la base de données afin d'extraire les motifs séquentiels. Le premier balayage vise à trouver les items fréquents F_1, le deuxième à trouver les séquences fréquentes de longueur 2 F_2[35], Le calcul de F_2 (les fréquents de taille 2) par S PADE, passe par une inversion de la base, qui la transforme d'un format vertical vers un format horizontal. Les auteurs considèrent que cette opération peut être simplifiée si la base peut-être chargée en mémoire vive [25].

Lors de la génération des séquences candidates, les jointures se font de la façon suivante [14]:

1. Élément contre élément : AB jointe avec AD en ABD.
2. Élément contre séquence : AB jointe avec A → C en AB → C.
3. Séquence contre séquence : A → B jointe avec A → C en A → BC,

A → B→ C et A→ C → B.

La procédure s'arrête quand aucune séquence fréquente ne peut être générée ou qu'aucune séquence ne peut être jointe. L'utilisation de bases de données verticales permet d'améliorer l'étape de vérification des séquences candidates [41].

```
     Données en entrée : DB, minsup
nr.  Produit la liste des séquences fréquentes
     Lire DB pour calculer F1, et les IdList(S) pour tout S ∈ F1
 1   k = 1
 2   while Fk ≠ ∅
 3   {
 4     Fk+1 = ∅
 5     forall P ∈ Fk, Q ∈ Fk if prefixe(P)=prefixe(Q)
 6     {
 7        Z = fusion(P, Q)
 8        Construct IdList(Z)
 9        if freq(Z) > minsup Fk+1 = Fk+1 ∪ {Z }
10     }
11     k = k + 1
12   }
13   return F1 ∪ F2 ... ∪ Fk
```

Figure 3.5. Le pseudo code de l'algorithme SPADE[42]

4.2.1.1. Limite de SPADE :

La nécessité d'une très grande mémoire pour transformer et puis après stocker toute la base de données.

4.3. Méthode par projection :

L'´equipe de l'Université Simon Fraser a proposé de partitionner la base de données initiale en fonction des préfixes des séquences. Cela permet une exécution parallèle et dirige l'effort d'extraction vers la partie utile de la base de données. Par exemple, si on ne souhaite connaitre que les séquences qui commencent par le préfixe ab, ces méthodes construiront la projection de la base initiale sur le préfixe ab, réduisant ainsi l'effort d'extraction en évitant de considérer la partie de la base qui n'est pas préfixée par ab [42]. L'utilisation de telles bases permet d'accélérer le comptage car la taille des bases projetées est réduite par rapport à la taille de la base initiale, chaque base étant plus facile à traiter. C'est le principe adopté par l'algorithme FreeSpan et amélioré par l'algorithme PrefixSpan.

4.3.1. L'algorithme FreeSpan :

FreeSpan (pour le terme anglais FREquEnt pattern-projected Sequential PAtterN mining) : est un algorithme proposé par Pei et al. En 2001 [40], est le premier algorithme qui considère la méthode de projection pour extraire des motifs séquentiels. Il trouve premièrement des itemsets fréquents et utilise ceux-ci pour construire des motifs séquentiels, dans le but de réduire la génération de séquences candidates [40]. FreeSpan utilise les items fréquents pour projeter récursive bases de données de séquence à un ensemble de plus petite projetée bases de données et des fragments sub-séquences dans chaque base de données projetée.

Le premier balayage de la base de données, recueille le support pour chaque item, et trouve l'ensemble items fréquents. Les items fréquents sont classés par ordre décroissant soutien (sous forme de article: support), par exemple, F_list = a: 4, b: 4, c: 4, d: 3, e: 3, f: 3.

Selon F_list, l'ensemble complet des motifs séquentiels en S peut être divisé en 6 sous-ensembles disjoints : (1) contenant seul item 'a', (2) contenant l'item 'b', mais ne contenant pas d'éléments après 'b' dans F_list, etc. des items non fréquents, tels que «g» dans cet exemple, sont retirés de la construction de bases de données projetées.

Notez que {b}, {c}, {d}, {e}, {f} bases de données projections sont construits simultanément de cours le premier balayage de la base de données de séquence initial [43].

Expérimental les résultats montrent que FreeSpan est efficace d'extraire l'ensemble des motifs et il est considérablement plus rapide que l'algorithme GSP. Le coût important de FreeSpan est de traiter avec des bases de données projetées [40].

4.3.2. L'algorithme PrefixSpan :

PrefixSpan (Prefix-projected Sequential pattern mining) Pei et al ont propose cet algorithme en 2001 [35], est un algorithme efficace pour l'extraction de séquences fréquentes qui s'appuie sur le même principe de bases projetées. Il fonctionne de manière récursive en réduisant l'espace de recherche à chaque étape, en évitant la génération de séquences non-fréquentes, Pour parvenir à cet objectif, PrefixSpan propose d'analyser les préfixes communs que présentent les séquences de données de la base à traiter. A partir de cette analyse, l'algorithme construit des bases de données intermédiaires qui sont des projections de la base d'origine déduites à partir des préfixes identifiés. Ensuite, dans chaque base obtenue, PrefixSpan applique un comptage du support des différents items afin de faire croître la taille des motifs séquentiels découverts [37].

On applique l'algorithme PrefixSpan sur la base de données suivante :

Client	Séquence
10	< (a) (a b c) (a c) (d) (c f) >
20	< (a d) (c) (b c) (a e) >
30	< (e f) (a b) (d f) (c) (b) >
40	< (e) (g) (a f) (c) (b) (c) >

Table 3.2: base de données exemple pour PrefixSpan.

En appliquant les étapes suivantes :

Étape 1 : Trouver les items fréquents. Pour cela, une passe sur la base de données va permettre de collecter le nombre de séquences supportant chaque item rencontré et donc d'évaluer le support des items de la base. Les items trouvés sont (sous la forme <item> : support) : <a> :4 ,<b> :4 ,<c> :4 ,<d> :3 ,<e> :3 ,<f> :3 .

Étape 2 : Diviser l'espace de recherche. L'espace de recherche complet peut être divisé en six sous-ensembles, puisqu'il y a six préfixes de taille 1 dans la base (i.e. les six items fréquents). Ces sous-ensembles seront : (1) les motifs séquentiels ayant pour préfixe <a>, (2) ceux ayant pour préfixe <b>, ... et (6) ceux ayant pour préfixe <f>.

Étape 3 : Trouver les sous-ensembles de motifs séquentiels. Les sous-ensembles de motifs séquentiels peuvent être trouvés en construisant les projections préfixées des bases obtenues et en réappliquant l'algorithme de fouille de manière récursive. Les bases ainsi projetées et les motifs obtenus sont alors en partie donnés à la table 3.3.

Préfixe	base projetée (suffixes)	motifs séquentiels
<a>	<(abc)(ac)(d)(cf)>, <(_d)(c)(bc)(ae)>, <(_b)(df)(c)(b)>, <(_f)(c)(b)(c)>	<a>, <(a)(a)>, <(a)(b)>, <(a)(bc)>, <(a)(bc)(a)>, <(a)(b)(a)>, <(a)(b)(c)>, <(ab)>, <(ab)(c)>, <(ab)(d)>, <(ab)(f)>, <(ab)(d)(c)>, <(a)(c)(a)>, <(a)(c)(b)>, <(a)(c)(c)>, <(a)(d)>, <(a)(d)(c)>, <(a)(f)>
<b>	<(_c)(ac)(d)(cf)>, <(_c)(ae)>, <(df)(c)(b)>, <c>	<b>, <(b)(a)>, <(b)(c)>, <(bc)>, <(bc)(a)>, <(b)(d)>, <(b)(d)(c)>, <(b)(f)>
⋮	⋮	⋮

Table 3.3 : Résultat de PrefixSpan sur la base de données de la table 3.2

Le nombre de bases de données intermédiaires est très important s'il y a beaucoup de séquences fréquentes. Si la base de données est grande, alors PrefixSpan nécessite une quantité importante de mémoire.

5. Conclusion :

Nous avons présenté dans ce chapitre une recherche les différentes méthodes d'extraction de motifs séquentiels. Parmi ces méthodes on a présenté des algorithmes baser sur Apriori comme GSP a été l'un de premières algorithmes, l'algorithme SPADE qui permet d'améliorer grandement le temps de calcul nécessaire ainsi qu'il augmenter les besoins de mémoire puisque il base sur la représentation vertical de la base de données, et en suite Les méthodes de recherche par projections sont plus efficaces pour l'extraction de motifs séquentiels. Les motifs séquentiels ont été utilisés pour analyser ces données et identifier les tendances, ont été utilisés pour mettre en œuvre des systèmes efficaces, aident à faire des prédictions, détecter des événements et dans l'aide générale à prendre des décisions stratégiques.

CHAPITRE 04

REALISATION

1. Introduction :

La réalisation est la dernière phase dans tout processus de développement d'un système ou d'un logiciel. Nous avons vu tous les préliminaires et concepts théoriques nécessaires à la bonne compréhension la méthode SPADE, et ceci pour bien appréhender notre travail. Dans ce chapitre, on vise de brièvement les outils et les moyens utilisé pour implémenter SPADE nous avons faire une petite comparaison avec l'algorithme GSP. Ainsi que l'environnement de programmation choisi, et l'ensemble des interfaces générés par notre application.

2. Implémentation :

Dans cette partie de mémoire nous voulons faire une petite comparaison entre l'algorithme GSP et l'algorithme SPADE pour bien déterminer la performance de cette dernière par rapport à la première et aussi leur point faible, pour cela nous présenter en détaille les étapes de leur exécution.

2.1. La fonctionnement de l'algorithme Generalized Sequential Pattern (GSP) :

Nous avons utilisé la base de séquence de la table en dessous.

SID	Temps(EID)	Séquence
1	10,15,20 ;25	<(CD)(ABC)(ABF)(ACDF)>
2	15,20	<(ABF)(E)>
3	10	<(ABF)>
4	10,20,25	<(DGH)(BF)(AGH)>

Table 4.1. Une base de données de séquence.

Pour déterminer les 1-séquence fréquents, nous considérons un support minimal= 2.

Le résultat est :

Items	Le support d'item
A	4
B	4
C	1
D	2
E	1
F	4
G	1
H	1

Table 4.2. Les candidats de 1-séquence.

Les items qui son support est supérieur ou égale le support minimal sont 1- Séquence fréquentes.

Séquence	**Le support séquence**
A	4
B	4
D	2
F	4

Table 4.3. Les 1-séquence fréquents par GSP

En joignant table 4.3 X table 4.3 mais la condition est k- 2 items doivent être communs.

séquence	**Le support de séquence**
A->A	1
A->B	1
A->D	1
A->F	1
AB	3
AD	1
AF	3
B->A	1
B->B	1
B->D	1
B->F	1
BD	4
BF	2
D->A	2
D->BD-F	2
D->D	1
DF	1
F->A	2
F->B	1
F->D	1
F->F	1

Table 4.4. Les candidats de 2-séquence par GSP

Les candidats qui satisfont au support minimal :

Séquence	**Le support**
AB	3
AF	3
B->A	2
BF	4
D->A	2
D->B	2
D->F	2
F->A	2

Table 4.5. les 2-séquence fréquents (GSP).

En joignant table 4.5 X table 4.5 pour obtenir des 3- séquence mais la condition est k- 2 items doivent être communs.

Séquence	**Le support de séquence**
ABF	3
BF->A	2
D->B->A	2
D->F->A	2
D->BF	2

Table 4.6. un 3-séquence fréquents(GSP)

Maintenant, en joignant table 4.6 X table 4.6 pour obtenir des 4- séquence.

Séquence	**Le support de séquence**
D->BF->A	2

Table 4.7. un 4-séquence fréquent.(GSP)

2.2. La fonctionnement de l'algorithme SPADE (Sequential Pattern Discovery using Equivalence classes) :

Est un algorithme utilisé pour la découverte rapide des motifs séquentiels. La plupart des algorithmes séquentiels d'extraction des motifs séquentiels mise en page de base de données horizontale .SPADE utilise le format de base de données verticale, où il charger le id -list sur disque pour chaque item, qui affiché dans figure ci-dessous

A		B		D		F	
SID	EID	SID	EID	SID	EID	SID	EID
1	15	1	15	1	10	1	20
1	20	1	20	1	25	1	25
1	25	2	15	4	10	2	15
2	15	3	10			3	10
3	10	4	20			4	20
4	25						

Table 4.8. ID_List des items fréquent.

La table 4.9 représente les items fréquents qu'ont un support >=2 le support minimal.

Items	**Le support d'items**
A	4
B	4
D	2
F	4

Table 4.9. Les 1-séquence fréquent (spade).

Dans l'étape suivante SPADE utilise la jointure temporelle pour identifier les séquences fréquentes. Pour calculer le soutien de la séquence AB, nous pouvons effectuer rejoindre temporelle (au même temps) ou non temporelle

AB		
SID	**EID A**	**EID B**
1	15	15
1	20	20
2	15	15
3	10	10

Table 4.10 id-liste 1-séquence temporelle

A→D		
SID	**EID A**	**EID D**
1	15	25
1	20	25

Table 4.11. id-liste 1-séquence non temporelle.

Et nous continuons de rejoindre tout les 1-séquence fréquent par non -temporelle et temporelle jointures pour obtenus la table au-dessous qu'est indiquant les 2-séquences fréquentes qui satisfissent le support minimal.

Séquence	**Le support de séquence**
AB	3
AF	3
BF	4
B→A	2
D→A	2
D→B	2
D→F	2
F→A	2

Table 4.11. les 2-séquence fréquent(SPADE).

Cette étape génère les 3-séquence :

ABF			
SID	**EID A**	**EID B**	**EID F**
	20	20	20
2	15	15	15
3	10	10	10

Table 4.12. id-liste2-séquence temporelle.

AB→A			
SID	**EID A**	**EID B**	**EID A**
1	15	15	20
1	20	20	25
2	15	15	-
3	10	10	-

Table 4.13. id-liste 2-séquence non temporelle.

Et nous continu la génération de même façon pour obtenu les 4-séquence présenté par la table 4.14 au-dessous

Séquence	Le support de séquence
ABF	3
AB→A	1
AF→A	1
BF→A	2
D→B→A	2
D→F→A	2
D→BF	2
F→AF	0
D→AF	1
D→AB	1
F→AB	0
B→AB	1
B→AF	1

Table 4.14. les 3-séquence résultant par (SPADE)

L'étape suivante est la génération les 3-séquence qui satisfissent le support minimal Comme le table ci-dessous :

ABF→A				
SID	**EID A**	**EID B**	**EID F**	**EID A**
1	20	20	20	25
2	15	15	15	-
3	10	10	10	-

Table 4.15. id liste de 4-séquence

D→BF→A				
SID	**EID D**	**EID B**	**EID F**	**EID A**
1	10	20	20	25
2	-	15	15	-
3	-	10	10	-
4	10	20	20	25

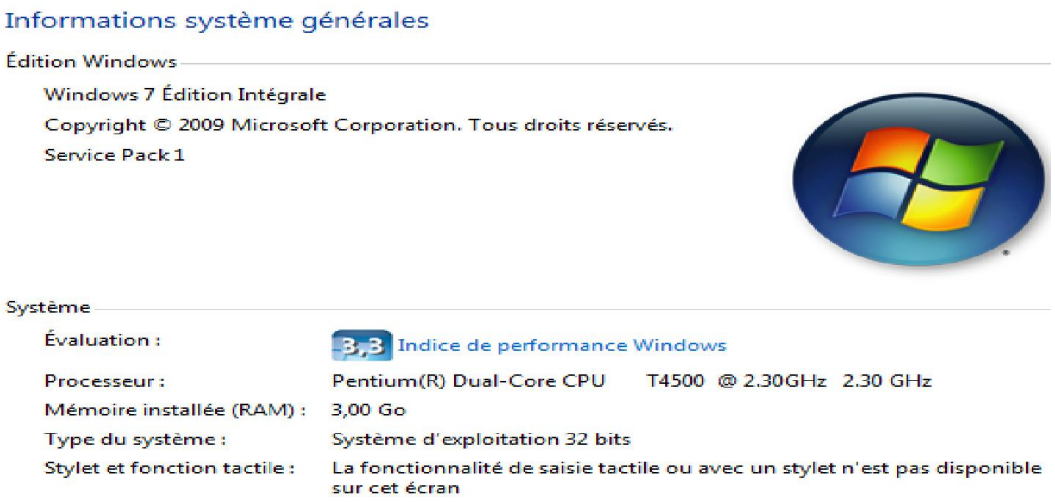
Table 4.16. id-liste de 4-séquence

Donc la séquence D ,BF,A a un support =2 qui satisfait le support minimal alors cette séquence est un 4-séquence fréquent.

Dans cette section, nous décrivons brièvement nos expériences dans l'application de l'extraction de la séquence, est une méthode efficace et évolutive pour l'extraction des motifs temporelle. Cette implémentation théorique se fait par une application qui permet de générer les k-séquence fréquent et extraire les motifs séquentiels rapidement que l'algorithme GSP

3. Environnement de l'application :

Notre application va être réalisée sur une machine qui comporte les caractéristiques suivant :



Environnement de développement

Pour mettre en œuvre notre application, nous avons choisis de travailler sur éclipse, Eclipse est une plateforme de développement écrite en Javaest un IDE (Integrated Development Environment) dédié au développement de logiciels basés sur Java (bien que d'autres langages soient supportés également).

- Java est aujourd'hui un langage aussi rapide que le c++ pourvu qu'on ne l'utilise pas pour une application très lourde (jeux en ligne, logiciel de traitement d'image, encodage vidéo etc...)
- Java est organisée, il contient des classes bien conçu et bien reparties.
- Java est connu et donc plus de chance de trouver des développeurs java; pour concevoir ou amélioré une application.
- L'avantage principal de Java par rapport aux autres langages c'est sa PORTABILITE, le fait qu'un programme Java puisse théoriquement être exécuté sur n'importe quelle plate-forme (type de processeur et système d'exploitation)

Enfin, nous notons que Eclipse nous a également fourni des classes prédéfinies et c'est un éditeur graphique permet de créer facilement l'interface graphique grâce à une barre d'outils permettant d'ajouter des composants à la fenêtre de l'application, et de modifier leurs propriétés.

4. L'architecture de l'application :

Nous avons utilisé la programmation orienté objet (POO) parce qu'il offre une lisibilité de manipulation du code source tel que maintenance, détection des erreurs et aider de corriger, débogage …etc.

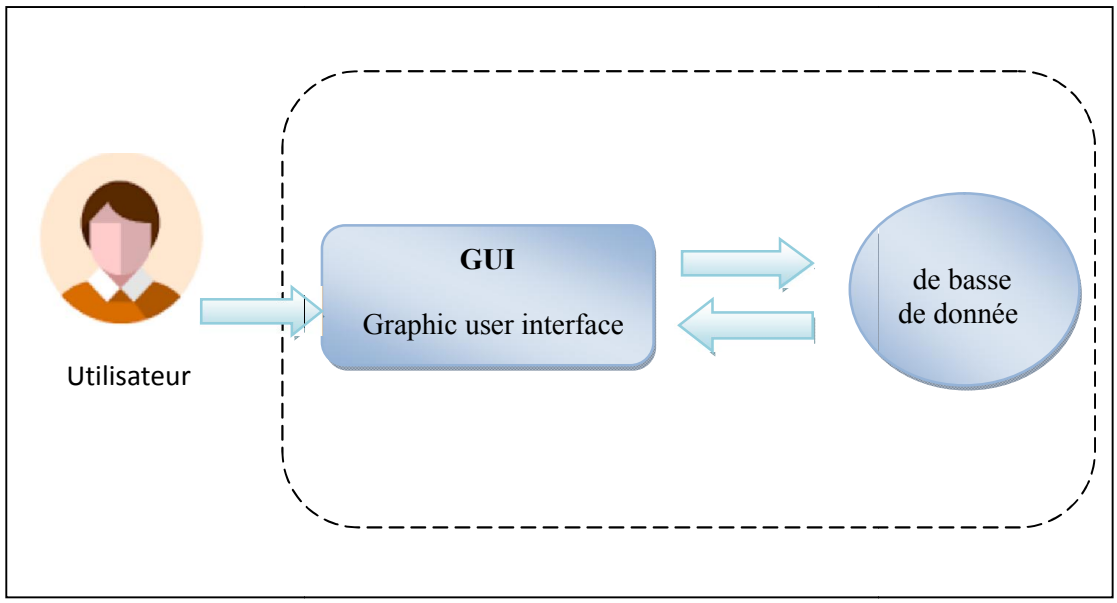


Figure 4.1. L'architecture de l'application.

- **L'interface graphique :**

C'est une simple interface graphique permet aux utilisateurs une meilleur interaction avec application. Nous avons essayé de donner un aspect éclipse et attrayant à travers une interface graphique ergonomique.

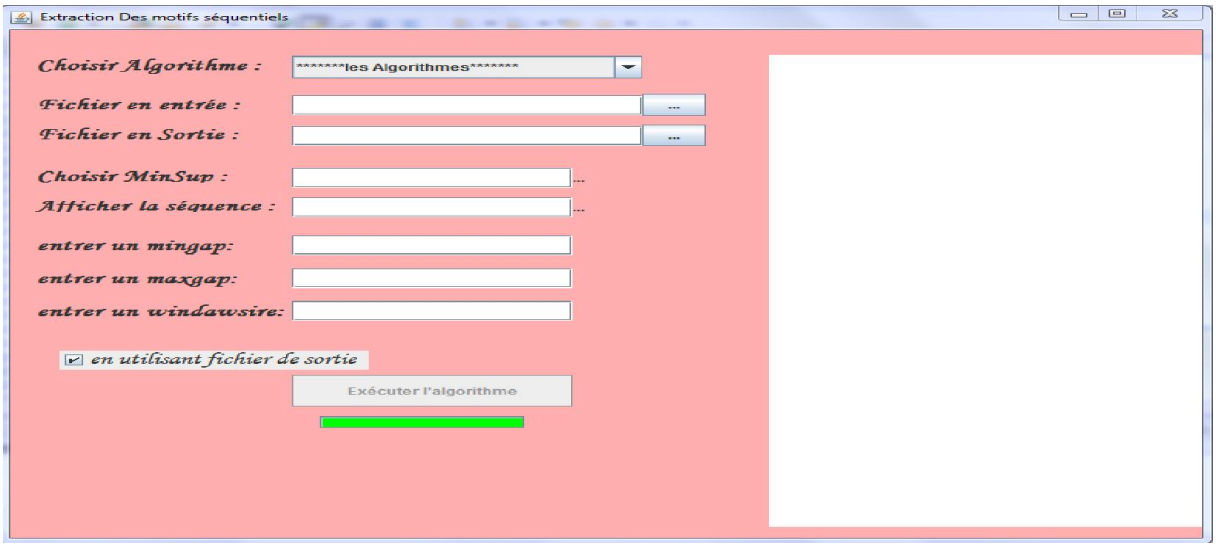
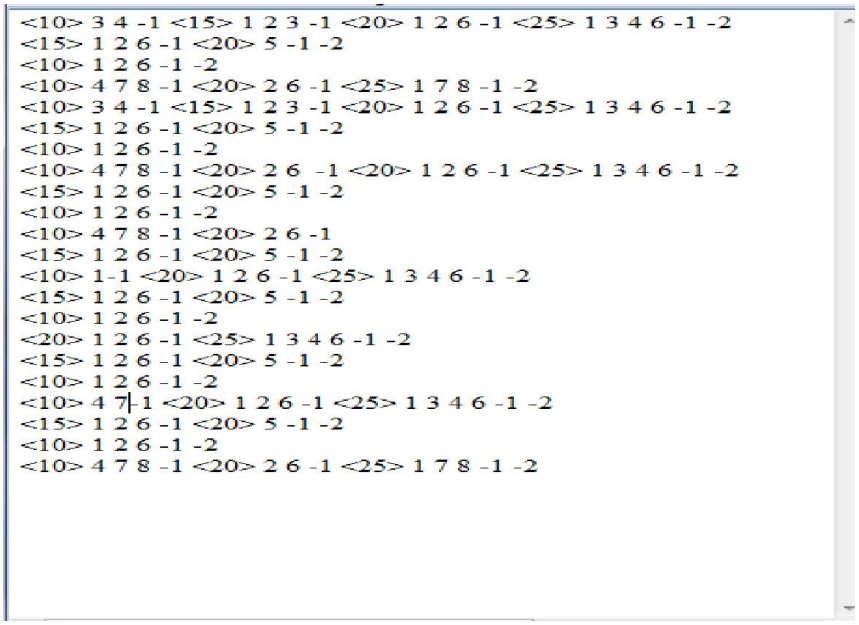


Figure 4.2 : interface graphique de l'application

- **La basse de donnée :** est un fichier texte contient une basse de donnée sous forme d'un texte chaque ligne est présenté une séquence et entre chaque transaction un séparateur (-1) et le (-2) indique que c'est la fin de séquence.



5. Conclusion :

Nous avons présenté SPADE, un nouvel algorithme pour l'extraction rapide de motifs séquentiels dans grandes bases de données. Contrairement aux approches GSP qui font plusieurs analyses de bases de données et utilisent structures de hachage arbre complexes, Nous avons montré comment on peut utilise de nouvelles stratégies d'élagage, qui peuvent être appliquées dans presque tous les domaines.

CONCLUSION GENERALE

Dans le cadre de ce travail de master, nous avons traité le problème de fouille de données (data mining) dans des bases de données. Ce type de problème est présent dans de nombreux domaines d'applications. Au cœur de ce mémoire nous avons présenté les différentes techniques de fouille de données (extraction de Connaissances) et Plus précisément la technique d'extraction des motifs séquentiels, pour mieux cerner la problématique posée, nous avons commencé par la présentation générale des notions concerne le processus d'extraction des connaissances à partir de données et leur sous-processus la fouille de donnée. En deuxième temps, nous avons passé à l'outil d'extraction des motifs fréquents(les règles d'association) on a parlé sur les concepts généraux et les algorithmes utiliser dans ce domaine et aussi nous avons présenté d'une manière générale les algorithmes d'extraction des motifs séquentiels .et puis nous avons implémenté un algorithme parmi ces algorithme. Nous avons choisi le langage java pour écrire et développé notre application, on a utilisé un fichier texte Contient base de données.

Comme perspective à ce travail, les derniers travaux contiennent d'ailleurs de plus en plus de contraintes sur la définition des motifs séquentiels. Initialement la recherche de motifs séquentiels est de considérer que données sont booléennes : un client achète ou n'achète pas un produit et pour minimiser l'espace de recherche, un objet ne peut intervenir qu'une seule fois dans un ensemble d'achats. Par contre, l'intérêt des motifs extraits est très discutable, quand nous obtiendrons des motifs de la forme :'les personnes qui ont acheté trois bouteilles de boisson ont aussi acheté deux fromages', les nombreuses valeurs numériques (acheter 2 ou 3 fromages est difficilement séparable strictement) rendent ces motifs difficiles à extraire et peu informatifs, et dans ce cadre, les experts travaillent pour assouplir ces notions, aussi étendre la théorie de la non-dérivabilité vers d'autres motifs tels que les arbres et les graphes.

Bibliographie

[1] G. Piatetsky-Shapiro, Data mining and knowledge discovery 1996 to 2005: overcoming the hype and moving from «university» to «business» and «analytics», Data mining and Knowledge Discovery.

[2] Alice Marascu, Extraction de motifs séquentiels dans les flux de données, Docteur en Sciences, France, 2009.

[3]Dhouha Grissa, Étude comportementale des mesures d'intérêt d'extraction de connaissances, doctorat, tunis, 2013.

[4] SASSI Amina, Une approche basée agent pour la fouille de données, magister en informatique, batna, 2013.

[5] René Lefébure et Gilles Venturi, Le Data Mining, Editions Eyrolles, 2001.

[6] Bouchekouf Asma, Perception du comportement de l'apprenant dans un environnement d'apprentissage, annaba, 2013.

[7] Jiawei Han and Micheline Kamber, Data mining concepts and techniques, 2nd edition , Diane Cerra San Francisco.

[8] Mohamed Hatem Haddad, Extraction et Impact des connaissances sur les performances des Systèmes de Recherche d'Information, docteur de l'université joseph fourier, 2002.

[9] Chami Djazia, Une plate forme orientée agent pour le data mining, magister, batna, 2010.

[10] Michael J. A. Berry, Gordon S. Linoff, Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management, chapitre 01, 2nd Edition, 2004.

[11] Lamiche Chaabane, fusion et fouille de donnees guidees par les connaissances: application a l'analyse d'image, doctorat, biskra, 2013.

[12] Tufféry, S., Data mining et statistique décisionnelle, Editions Technip. (2007).

[13] Benamar houmadi, étude exploratoire d'outils pour le data mining, doctorat, l'université du québec a trois-rivières, 2007.

[14] Guillaume Calas, Études des principaux algorithmes de data mining, Spécialisation Sciences Cognitives et Informatique Avancée, France, 2009.

[15] les arbres de décision
http://www.grappa.univlille3.fr/polys/apprentissage/sortie004.html
Consulté le 12/03/2016.

[16] Kellou Kenza et Mokhtari Abdeldjalil, Réalisation d'une plateforme d'expérimentations et de tests d'algorithmes de data mining, magister,
[17] Les plus proches voisins
http://www.grappa.univlille3.fr/polys/fouille/main.tgz
Consulté le 13/03/2016.
[18] C. Scharff, Méthode des k plus proches voisins, IFI, 2004.
[19] Philippe Preux. « Fouille de données. Notes de cours ». Université de Lille. 31 août 2009.
[20] Mohamed El hadi Benelhadj, entrepôt de données et fouille de données un modèle binaire et arborescent dans le processus de génération des règles d'association, doctorat, constantine
[21] Daniel Rajaonasyfeno, Mesures de qualité des règles d'association : normalisation et caractérisation des bases, doctorat, France, 2007.
[22] Sarra Ayouni, Etude et Extraction de Règles graduelles floues : Définition d'algorithmes efficaces, doctorat, tunis, 2012.
[23] Hassane Hilali, application de la classification textuelle pour l'extraction des règles d'association maximales, université du québec, 2009.
[24] Nicolas Pasquier, Data Mining : algorithmes d'extraction et de réduction des règles d'association dans les bases de données, doctorat, France, 2000.
[25] Allia Mohamed Rachid, BOUADI Tassadit, El MOUTAOUKIL Sami, et KEIRA Mamadou, Fouille de données : Règles séquentielles, master.
[26] Ansaf Salleb, Recherche de motifs fréquents pour l'extraction de règles d'association et de caractérisation, doctorat, Orléans, 2003.
[27] Abdelhak Mansoul, fouille de données biologiques : étude comparative et expérimentation, magister, oran, 2010.
[28] gwenael bothorel, algorithmes automatiques pour la fouille de données visuelle et la visualisation de règles d'association. Application aux données aéronautiques, doctorat, Toulouse, 2014.
[29] Alouan Basma, recherche de partitions floues optimale par la segmentation floue pour la fouille de données quantitatives, magister, boumerdes, 2008.
[30] Rahmani Rabah, découvret d'association sémantique dans les bases de données relationnelles par des méthodes de data mining, magister, tizi-ouzou.
[31] Jérôme Azé, Extraction de connaissances à partir de données numériques et textuelles, doctorat, France, 2003.

[32] Bilal Idiri, Méthodologie d'extraction de connaissances spatio-temporelles par fouille de données pour l'analyse de comportements à risques - Application à la surveillance maritime, doctorat, paris, 2013.

[33] Mickaël Fabrègue, Extraction d'informations synthétiques à partir de données séquentielles Application à l'évaluation de la qualité des rivières, doctorat, strasbourg, 2014.

[34] M'zali hassen, les règles d'association séquentielles, magister, 2006.

[35] Elias Egho, Extraction de motifs séquentiels dans des données séquentielles multidimensionnelles et hétérogènes Une application à l'analyse de trajectoires de patients, doctorat, Lorraine, 2014.

[36] Julien Rbatel, extraction de motifs contextuels : Enjeux et application dans les données séquentielles, France, 2011.

[37] Chedy Raïssi, Extraction de séquences fréquentes : des bases de données statiques aux flots de données, Montpellier, 2008.

[38] Asma Ben Zakour, Extraction des utilisations typiques à partir de données hétérogènes historiées en vue d'optimiser la maintenance d'une flotte de véhicules, doctorat, Bordeaux, 2012

[39] Maguelonne Teisseire, Autour et alentours des motifs séquentiels, doctorat, 2007.

[40] Thabet Slimani, Amor Lazzez, sequential mining: patterns and algorithms analysis, Computer Science, Taif University & LARODEC Lab, Saudia Arabia

[41]Marc Plantevit, Extraction De Motifs Séquentiels Dans Des Données Multidimensionnelles, doctorat, France, 2008.

[42] Hunor albert-lorincz, Contributions aux techniques de Prise de Décision et de Valorisation Financière, doctorat, lyon, 2007.

[43] Manish Gupta, Jiawei Han, Approaches for Pattern Discovery Using Sequential Data Mining, Université de Illinois at Urbana-Champaign, USA.

الملخص

استخراج النماذج المتسلسلة هي تقنية ذات أهمية في مجال استخراج المعلومات من قواعد البيانات، و هي تستعمل في كثير من المجالات وخاصة في مجال تحليل المعلومات في المجمعات الخاصة بالمبيعات.

مشكلة اكتشاف الأنماط المتسلسلة يرتكز على قاعدة بيانات للمعاملات بحيث هده المعاملات هي قائمة سلع متعلقة بالوقت او الزمن. وهدا المجال هو الأكثر صعوبة من مجال استخراج قواعد الترابط . يطبق عدة خوارزميات للحصول على أفضل النتائج بالنسبة إلى زمن التنفيذ وتقليل مساحة مجال البحث

الكلمات المفتاحية: استخراج البيانات، و قواعد البيانات, قواعد الترابط ، و نماذج متسلسلة .

Abstract

The extraction of sequential patterns is a significant challenge for data mining community, they are involved in areas more and more especially in the sales data analysis business organizations , the problem of discovery sequential patterns is given a transactional database where transactions are lists of items with time constraints . It is the most difficult area of discovery of sequential patterns, compared with research association rules. The domain uses multiple algorithm achieved a better result in terms of execution time and minimize the motives search space.

Keywords : data mining, databases , association rules , sequential patterns .

Résumé

L’extraction de motifs séquentiels est un défi important pour la communauté fouille de données, ils se trouvent impliqués dans des domaines de plus en plus nombreux en particulier dans l’analyse de données de vente d’organisations commerciales, Le problème de la découverte des motifs séquentiels consiste, étant donné une base de données transactionnels où les transactions sont des listes d’items avec des contraintes de temps. Il est le plus difficile du domaine de la découverte des motifs séquentiels, comparativement à la recherche des règles d’association. Ce domaine utilise plusieurs algorithme pour obtenue un meilleur résultat en terme de temps d’exécution et minimiser l’espace de recherche des motifs.

Mots clé : la fouille de donnée, les bases de données les règles d’association, les motifs séquentiels.