REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

Faculté des Mathématiques et de l'Informatique
Département d'Informatique
N° : ……

DOMAINE : Mathématiques et Informatique
FILIERE : Informatique
OPTION : RESEAOUX ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

Mémoire présenté pour l'obtention
Du diplôme de Master Académique

Par : Zidane Soheyb
Mahdi Omer Ziad

Intitulé

An open-source software and data mining library written in PHP ,specialized in pattern mining

Soutenu devant le jury composé de :

Bounif Mohamed	Université de M'sila	Président
KamelEddine HERAGUEMI	Université de M'sila	Rapporteur
Mezrag Fares	Université de M'sila	Examinateur

Année universitaire : 2021 / 2022

REMERCIEMENTS

Après avoir rendu grâce à Allah le tout puissant et le miséricordieux, nous tenons à remercier :

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait à pu avoir le jour sans l'aide et l'encadrement de Mr Kamel Eddine HERAGUEMI, on le remercie pour la qualité de son encadrement, pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce mémoire.

Notre remerciement s'adresse également à tous nos professeurs pour leurs générosités et la grande patience dont ils ont su faire preuve malgré leurs charges académiques et professionnelles

DÉDICACES

Je dédie entièrement ce travail à mon père et à ma mère, mes piliers, mes exemples, mes premiers supporteurs et ma plus grande force. Merci pour votre présence, votre soutien, votre aide financière, et surtout votre amour, merci de n'avoir jamais douté de moi.

Tout ce que j'espère, c'est que vous soyez fiers de moi aujourd'hui.

, qui font de mon univers une merveille, je leurs souhaite beaucoup de bonheur et de réussite.

A ma collègue qui depuis des années m'encourage, me comprend et a toujours été à mes côtés, que dieu lui donne du bonheur, santé et réussite.

A tous les, les voisins et les amis que j'ai connu jusqu'à maintenant. Merci pour leurs amours et leurs encouragements.

Introduction générale :

Le volume des données numériques collectées et conservées au niveau des systèmes d'information d'aujourd'hui ne cesse de croître et le besoin d'extraire des informations utiles à partir de ces données ne cesse de se transformer en une stratégie. Les entreprises comprennent, maintenant, que leurs données ne sont plus utiles pour uniquement l'utilisation fonctionnelle classique connue, mais qu'elles peuvent leur trouver des utilisations encore plus avancées. Les énormes masses de données conservées souvent dans de gigantesques bases de données dormantes, peuvent sans aucun doute contenir des connaissances de très grande valeur commerciale ou scientifique n'attendant qu'à être exploitées.

La question qui se pose est alors : Comment peut-on extraire les informations cachées au sein de grandes bases de données ? La puissance de calcul des ordinateurs actuels et la baisse des coûts de stockage laissent prédire que nous disposons des moyens physiques pour le faire. Le problème réside alors au niveau logiciel. En effet, les bases de données classiques ne sont plus de taille à faire face à l'analyse de telles informations et c'est grâce à ce besoin pressant que sont apparues les techniques d'extraction de connaissances à partir des données communément connues sous le nom de « Data Mining ». Citons parmi ces techniques : les règles d'association, l'analyse de liens, la détection de clusters, les algorithmes génétiques, les arbres de décision et les règles d'association séquentielles.

Dans La technique des règles d'association est un moyen très répandu qui permet de rechercher et détecter les liens (associations) entre les données. Le principe de cette technique est simple à implémenter ; il consiste à extraire des connaissances en passant par deux étapes:

- l'extraction de motifs intéressants (fréquents).
- génération de règles d'association à partir de ces motifs.

Un motif décrit des relations dans un sous-ensemble de données avec une certaine certitude.

Selon une mesure d'intérêt choisie par l'utilisateur, un motif intéressant et appelé connaissance si cette mesure d'intérêt est relativement supérieure à un seuil fixé a priori.

Notre travail consiste dans un premier temps à étudier et à comprendre le fonctionnement des algorithmes d'extraction dans un deuxième temps, à évaluer et à comparer les performances de ces algorithmes en fonction de différents langages JAVA et PHP.

La structure de ce mémoire va comme suit :

- Le premier chapitre présente les fondamentaux du data Mining

Introduction générale :

- Dans le deuxième chapitre, nous expliquerons les Algorithmes EXACTE de règles d'association.
- Le troisième chapitre est consacré à la présentation des outils utilisés pour l'implémentation de noter algorithmes et les diagrammes de classe.
- Le quatrième chapitre Il présente également les expérimentations, les résultats et une discussion sur ces résultats.

Et pour finir, nous conclurons tout en proposant des perspectives.

TABLE DES MATIERES

Liste des Tableaux

Liste des figures

CHAPITRE 01 : EXPLORATION DE DONNÉES ET EXPLORATION DE RÈGLES D'ASSOCIATION

1.1Introduction

L'exploration de données, connue aussi sous l'expression de fouille de données, forage de données, prospection de données, data mining, ou encore extraction de connaissances à partir de données à l'aide de méthodes automatiques ou semi-automatiques utilisant un ensemble d'algorithmes de disciplines scientifiques diverses telles que les statistiques, l'intelligence artificielle ou l'informatique, pour construire des modèles à partir des données, c'est-à-dire trouver des structures intéressantes ou des motifs selon des critères fixés au préalable, et d'en extraire un maximum de connaissances utiles pour l'entreprise.[1]

1.2 Extraction de connaissance à partir de données

Le processus d'extraction de l'information consiste à parcourir les données volumineuses contenues dans une base de donnée. Ce processus est décrit dans le schéma suivant. (Figure1.1). Ce processus comprend des étapes de définition du problème (définition du domaine, but de l'utilisateur final), de préparation des données (sélection, Préparation, transformation), de fouille de données (sélection, des outils de data mining appropriés, recherche des patrons) et d'évaluation des résultats pour aboutir aux nouvelles connaissances. Le processus présenté est itératif et plusieurs retours en arrière dans les différentes étapes peuvent être nécessaires pour affiner les résultats [2].

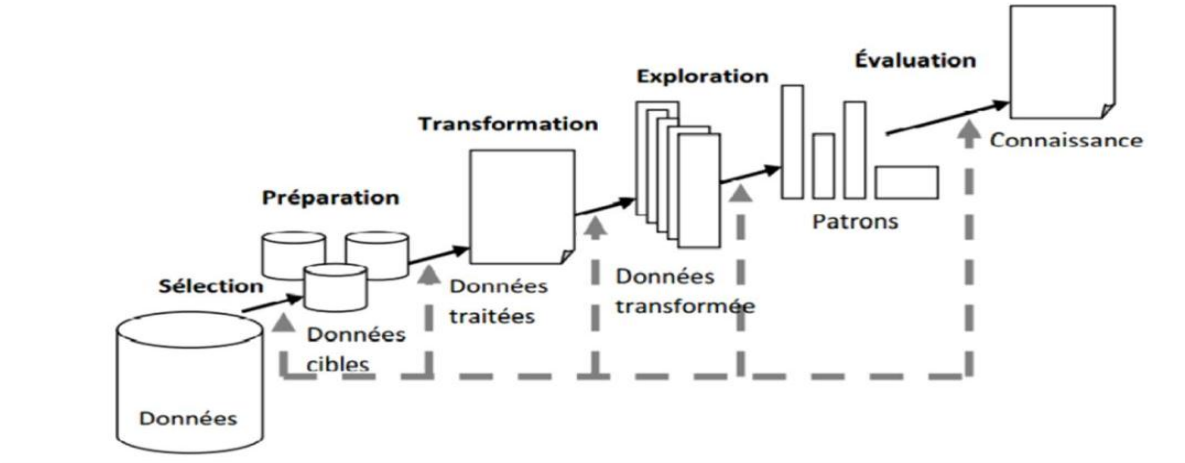

Figure1.1 Processus de découverte de connaissances à partir de données [20].

- La sélection qui crée un ensemble de données à étudier
- Le prétraitement qui vise à enlever le bruit et à définir une stratégie pour traiter les données manquantes.
- La transformation où l'on recherche les meilleures structures pour représenter les données en fonction de la tâche.
- L'Exploration: la fouille proprement dite et la définition de la tâche : classification, recherche de modèles…et la définition des paramètres appropriés.
- Évaluation et l'évaluation pendant laquelle les patrons extraits sont analysés. La connaissance qui en est ainsi extraite est alors stockée dans la base de connaissances.

1.3 Définition data Mining (Fouille de données) :

Data Mining signifie littéralement « fouille de données » ou « forage de données ». Ce procédé, basé sur une série d'algorithmes ou modèles de data Mining permet d'extraire des informations à partir de données, informations qui, grâce à l'analyse, se convertissent en connaissances. Le data mining est l'analyse d'un ensemble d'observations qui a pour but de trouver des relations insoupçonnées et résumer les données d'une nouvelle manière, de façon qu'elles soient plus compréhensibles et utiles pour leurs détenteurs ». (David Hand, 2001). Autrement dit, il consiste à analyser des informations collectées dans des entrepôts de données afin d'y détecter des relations qu'il serait a priori impossible d'identifier sans cet outil. C'est un élément essentiel dans la relation client et de système d'aide à la décision. Le Data Mining est un processus inductif, itératif et interactif dont l'objectif est la découverte de modèles de données valides, nouveaux, utiles et compréhensibles dans de larges bases de données [3].

1.4 Tâches du DataMining

Le DataMining est utilisé pour accomplir plusieurs tâches :

1.4.1. Association Rule mining (règle d'association)

Cette fonction du datamining permet de découvrir quelles variables vont ensemble, quelles sont les règles qui vont permettre de quantifier les relations entre deux ou plusieurs variables. Par exemple, si l'on s'intéresse à 500 clients qui viennent faire leurs courses au supermarché le vendredi soir et que l'on constate que sur ces 500 clients, 100 achètent des fruits et que sur ce nombre, 30 achètent du lait, ainsi, la règle d'association est « Si L'on achète des fruits,

alors on achète du lait », avec une mesure de support de 100/500 = 20% et un seuil de confiance de 30/100 = 33% [4].

1.4.2 Segmentation (analyse des clusters)

La variable cible n'est pas numérique, mais catégorielle, comme par exemple le revenu, qui peut être divisé en trois catégories : faible revenu, revenu moyen et revenu élevé. « La segmentation consiste à répartir les clients en groupes homogènes, qu'il Convient ensuite d'aborder par des moyens spécifiques et adaptés aux caractéristiques et attentes de chaque groupe. Les membres d'un même groupe réagissent de la même manière aux stimuli marketing. Ils ont en commun un mode de communication, des Comportements d'achat et/ou des besoins spécifiques » [4].

1.4.3 Classification

La classification est la tâche la plus commune du Data Mining et qui semble être une obligation humaine. Afin de comprendre notre vie quotidienne, nous sommes constamment classifiés, catégorisés et évalués. La classification consiste à étudier les caractéristiques d'un nouvel objet pour lui attribuer une classe prédéfinie. Les objets à classifiés sont généralement des enregistrements d'une base de données, la classification consiste à mettre à jour chaque enregistrement en déterminant un champ de classe. La tâche de classification est caractérisée par une définition de classes bien précise et un ensemble d'exemples classés auparavant. L'objectif est de créer un modèle qui peut être appliqué aux données non classifiées dans le but de les classifiées [5].

1.5 Règles d'association

1.5.1 Définition

Dans le domaine du data mining la recherche des règles d'association est une méthode populaire étudiée d'une manière approfondie dont le but est de découvrir des relations ayant un intérêt pour le statisticien entre deux ou plusieurs variables stockées dans de très importantes bases de données. Piatetsky-Shapiro1 présentent des règles d'association extrêmement fortes découvertes dans des bases de données en utilisant différentes mesures d'intérêt. En se basant sur le concept de relations fortes, Rakesh Agrawal et son équipe présente des règles d'association dont le but est de découvrir des similitudes entre des produits dans des données saisies sur une grande échelle dans les systèmes informatiques des points de ventes des chaînes de supermarchés. [6]

Exemple de règle d'association :

- Si un client achète du lait alors il achète du pain (90%) - Si un client achète une télévision, il achètera un récepteur satellite dans un mois (50%) - Si maladie X et traitement Y alors guérison (95%) - Si maladie X et traitement Y alors guérison dans Z années (97%) - Si présence et travail alors réussite à l'examen (99%) Ces règles sont intuitivement faciles à interpréter car elles montrent comment des produits ou des services se situent les uns par rapport aux autres. Elles sont particulièrement utiles en marketing et peuvent être facilement utilisées dans le système d'information de l'entreprise. Le but principal de cette technique est donc descriptif. Dans la mesure où les résultats peuvent être situés dans le temps, cette Technique peut être considérée comme prédictive. Cependant, il faut noter que cette méthode, si elle peut produire des règles intéressantes, peut aussi produire des règles triviales ou inutiles (provenant de particularités de l'ensemble d'apprentissage). La recherche de règles d'association est une méthode non supervisée car on ne dispose en entrée que de la description des achats.[**6**] [7].

1.6 Règles d'association mesures

Comme mentionné ci-dessus, l'objectif du processus KDD est d'extraire des modèles de données qui devraient être valides et compressibles pour l'utilisateur. En règle générale, les utilisateurs ne savent pas quelles connaissances seront générées à partir de ces données. Par conséquent, les modèles intéressants doivent être quantifiés par certaines mesures qui définissent son importance pour l'utilisateur. Sur la base de l'interaction de l'utilisateur avec le processus KDD, les mesures d'intérêt dans ARM pourraient être divisées en deux catégories principales : les mesures subjectives et objectives. Le premier prend en compte à la fois les buts (objectifs) de l'utilisateur et les mesures des données, tandis que le second est basé uniquement sur les informations des données. Dans ce qui suit, nous avons détaillé les mesures objectives importantes dans ce domaine, qui ne nécessitent aucune expérience de l'utilisateur. [7]

1.6.1. Définition Support :

Le support est la proportion de transactions dans D qui contient X, par rapport au total des enregistrements dans la base de données.

Le support est calculé à l'aide de l'équation suivante :

$$\text{Support}(X) = \text{Nombre de transactions contenant } X \,/\, \text{Nombre total de transactions}$$

Le soutien d'une règle d'association $X \Rightarrow Y$ est le soutien de $X \cup Y$.

1.6.2. Définition Confiance :

La confiance est la proportion de transactions couvrant X et Y, par rapport au total des enregistrements contenant X. Lorsque le pourcentage dépasse un seuil de confiance, une règle d'association intéressante peut être générée. La confiance de la réglée calculée comme suit :

$$\text{Confiance } (X \Rightarrow Y) = \text{support}(X \cup Y) / \text{support}(X)$$

La confiance est extraite de la force. Une règle d'association $X \rightarrow Y$

avec une confiance de 80%, cela signifie que 80% des transactions contenant X contiennent également Y. Sur la base de ces deux mesures (Support et Confiance), ARM vise à extraire toutes les règles qui satisfont à la fois le support minimum spécifié par l'utilisateur et les seuils de confiance minimum. Sans nervosité, ces mesures ne pouvaient évaluer l'intérêt réel et la qualité des règles extraites, par conséquent, une grande variété de propositions définissent d'autres mesures de qualité pour faire face à la faiblesse du cadre de soutien - confiance telles que: Levée, Effet de levier et Conviction, etc. dans ce qui suit, nous détaillons et définissons les mesures objectives les plus fréquemment utilisées ARM question. [7]

$$\text{Lift } (A \Rightarrow C) = \frac{\text{Confiance } (A \Rightarrow C)}{\text{Support}(C)}$$

En d'autres termes, la mesure lift quantifie le degré de dépendance entre l'antécédent et la conséquence de la règle. Plus précisément, si l'ascenseur est égal à 1, l'antécédent et la conséquence sont indépendants; sinon, s'il est supérieur à 1, l'antécédent et la conséquence sont positivement dépendants. Sinon, ils sont négativement dépendants. Par conséquent, la plupart des propositions recherchent la dépendance positive entre les deux parties de la règle.

1.6.3. Définition Leverage :

La mesure de levier (nouveauté) quantifie à quel point la Co - occurrence de l'antécédent et de la conséquence est différente. L'effet de levier de la règle est calculé comme suit:

$$\text{Leverage } (A \Rightarrow C) = \text{support}(A \Rightarrow C) - (\text{support}(A) \times \text{support}(C))$$

Généralement, l'effet de levier prend ses valeurs dans la plage de [-0,25, 0,25], où une valeur fortement positive indique une forte association entre A et C, tandis que les valeurs négatives indiquent une faible association entre A et B et une forte association entre A et C. En plus de celles-ci,

diverses mesures sont proposées pour faire face à d'autres faiblesses de la mesure de soutien-confiance. [7]

1.6.4. Définition Conviction :

La conviction a d'abord été proposée par Brain et al. En 1997. Il s'inspire de la définition logique de l'implication et représente le degré d'implication de la règle. La conviction varie le long des valeurs (0,5,..,∞,) lorsque ses valeurs sont éloignées de 1, cela indique l'intérêt de la règle. Contrairement à d'autres mesures, il est sensible à la direction de la règle, signifie conv $(A \Rightarrow C) \neq$ conv $(C \Rightarrow A)$.Il est défini comme :

$$\text{Conviction}\,(A \Rightarrow C) \;=\; {}^{1-\text{support}\,(A)\;/}\quad 1-\text{support}\,(A \Rightarrow C)$$

Où, |C | et |A ∪ C| sont le nombre d'éléments dans la partie Conséquence et la règle total respectivement. La compréhension augmente et les règles sont plus claires sensibles chaque fois que le nombre d'éléments dans la partie antérieure est plus petit.

De plus, il est important d'extraire les règles qui ont un faible support dans l'ensemble de la base de données mais de telles règles rares peuvent être intéressantes pour les décideurs. Ainsi, une autre mesure est définie pour ce problème appelée intérêt. [7]

1.6.5. Définition L'intérêt:

L'intérêt d'une règle est utilisé pour quantifier à quel point la règle est surprenante pour les utilisateurs. Comme le point le plus important de l'exploration de règles est de trouver des informations cachées, il devrait découvrir ces règles ayant relativement moins d'occurrences dans la base de données. La mesure de l'intérêt est définie par :

$$\text{Intéressant}\,(A \Rightarrow C) = \frac{\text{Supp}\,(A \cup C)}{\text{Supp}\,(A)} \times \frac{\text{Supp}\,(A \cup C)}{\text{Supp}(C)} \times \frac{(1-\text{Supp}\,(A \cup C)}{N}$$

Où A, C et N sont respectivement l'antécédent, la conséquence et le nombre de transactions dans l'ensemble de la base de données.

Exemple :

Considérons la base de données de transactions modélisée dans le tableau 2.5, qui contient 5 transactions t1,t2,t3,t4,t5 et 5 éléments A,B,C,D, E. Par exemple, pour calculer le support de l'ensemble de 2 éléments A, B, nous devons calculer le nombre de transactions qui comprend à la fois A et B, égal à 2, ainsi, le support de A , B est 2/5. Par la suite, la confiance de la règle A ⇒ B est égale à proportion du support (A ⇒ B) au support(A), soit = 2/3. Si nous considérons que Min-sup et Min-conf sont égaux à 2/5 et 2/3, respectivement, alors la règle A ⇒ B est acceptée. [7]

t1	A	B	C
t2	A	B	
t3	C	D	
t4	E	D	
t5	C	A	

TABLEAU 1.1: Un exemple de base de données transactionnelle

De plus, les autres mesures de la règle A ⇒ B sont calculées comme suit:

- *Lift (*A⇒B*) =5/3.*
- *Leverage (*A⇒B*) =4/25.*
- *Conviction (*A ⇒B*)=9/5*
- *Compréhensibilité (*A⇒B*)* = log (2) / log (3) = 0.63.
- *Intéressant (*A⇒B*)* $= \frac{2/5}{3/5} \times \frac{2/5}{2/5} \times \frac{1-2/5}{5}$ = 2/25.

Conclusion:

Nous avons discuté des règles d'association dans l'exploration de données

- À propos des règles d'association dans l'exploration de données
- Règles de fonctionnement de l'association
- Algorithmes dans les règles d'association
- Utilisations des règles d'association

CHAPITRE 02 :
LES ALGORITHMES DES RÈGLES D'ASSOCIATION

2.1. Introduction :

L'extraction de règles d'association et de bi-clusters sont deux techniques de fouille de données complémentaires majeures, notamment pour l'intégration de connaissances. Ces techniques sont utilisées dans de nombreux domaines, mais aucune approche permettant de les unifier n'a été proposée. Hors, réaliser ces extractions indépendamment pose les problèmes des ressources nécessaires (mémoire, temps d'exécution et accès aux données) et de l'unification des résultats. Nous proposons une approche originale pour extraire différentes catégories de modèles de connaissances tout en utilisant un minimum de ressources. Cette approche est basée sur la théorie des ensembles fermés et utilise une nouvelle structure de données pour extraire des représentations conceptuelles minimales de règles d'association, bi-clusters et règles de classification. Ces modèles étendent les règles d'association et de classification et les bi-clusters classiques, les listes d'objets supportant chaque modèle et les relations hiérarchiques entre modèles étant également extraits. Cette approche a été appliquée pour l'analyse de données d'interaction protéiniques entre le virus VIH-1 et l'homme. L'analyse de ces interactions entre espèces est un défi majeur récent en bio-informatique. Plusieurs bases de données intégrant des informations hétérogènes sur les interactions et des connaissances biologiques sur les protéines ont été construites. Les résultats expérimentaux montrent que l'approche proposée peut traiter efficacement ces bases de données et que les modèles conceptuels extraits peuvent aider à la compréhension et à l'analyse de la nature des relations entre les protéines interagissant.

2.2. Les Algorithmes EXACTE

2.2.1 Algorithme Apriori

L'algorithme Apriori est une séquence d'étapes à suivre pour trouver le jeu d'éléments le plus fréquent dans la base de données donnée. Cette technique d'exploration de données suit les étapes de jointure et d'élagage de manière itérative jusqu'à ce que le jeu d'éléments le plus fréquent soit atteint. Un seuil de support minimum est donné dans le problème ou il est supposé par l'utilisateur.

1- Dans la première itération de l'algorithme, chaque élément est pris comme un candidat à 1 itemsets. L'algorithme comptera les occurrences de chaque élément.
2- Qu'il y ait un support minimum, min_sup (par exemple 2). L'ensemble des ensembles de 1 items dont l'occurrence satisfait le min sup est déterminé. Seuls les candidats dont

le nombre est supérieur ou égal à min_sup sont pris en compte pour l'itération suivante et les autres sont élagués.

3- Ensuite, les éléments fréquents de 2-itemset avec min_sup sont découverts. Pour cela, à l'étape de jointure, le jeu de 2 éléments est généré en formant un groupe de 2 en combinant des éléments avec lui-même.
4- Les candidats à 2 items sont élagués à l'aide de la valeur de seuil min-sup. Maintenant, la table aura 2 ensembles d'éléments avec min-sup uniquement.
5- La prochaine itération formera des ensembles de 3 éléments en utilisant l'étape de jointure et d'élagage. Cette itération suivra la propriété anti monotone où les sous-ensembles de 3 ensembles d'éléments, c'est –à-dire les sous-ensembles de 2 ensembles d'éléments de chaque groupe tombent dans min_sup. Si tous les sous-ensembles de 2 items sont fréquents, le sur-ensemble sera fréquent sinon il est élagué.
6- L'étape suivante suivra la création d'un ensemble de 4 éléments en joignant un ensemble de 3 éléments avec lui-même et en élaguant si son sous-ensemble ne répond pas aux critères min_sup. L'algorithme est arrêté lorsque le jeu d'éléments le plus fréquent est atteint.

2.2.2 L'Algorithme d'Apriori: Pseudo Code

Input : Ck: i$temsets$ $candidats$ de $taille$ k.

$Output$: Lk: i$temsets$ $fréquents$ de $taille$ k

$L1$ = {i$tems$ $fréquekts$};

for (k = 1; Lk! = ∅; k + +) do

Ck + 1 = $candidats$ $générés$ à $partir$ de L ;

$Pour$ $chaque$ $transaction$ t de la $base$ de données, in$crémenter$ $lecompteur$ de $tous$ les $candidat$

+ 1qui $sont$ $contenus$ $dans$ t.

Lk + 1 = $candidats$ $dans$ Ck + 1$avec$ $Min Supp$.

$Return$ ∪ k Lk.

2.2.3 Exemple d'Apriori :

Seuil de support=50%, Confiance= 60%

Transaction	Liste d'objets
T1	I1, I2, I3
T2	I2, I3, I4
T3	I4, I5
T4	I1, I2, I4
T5	I1, I2, I3, I5
T6	I1, I2, I3, I4

Solution:

Seuil de support = 50% => 0,5 * 6 = 3 => min_sup = 3

TABLEAU 2.1. Nombre de chaque article

Article	Compter
I1	4
I2	5
I3	4
I4	4
I5	deux

TABLEAU 2.2 Étape de taille: montre que l'élément I5 ne répond pas à min_sup = 3, il est donc supprimé, seuls I1, I2, I3, I4 rencontrent le nombre min_sup.

Article	Compter
I1	4
I2	5
I3	4
I4	4

TABLEAU 2.3. Rejoignez l'étape: Ensemble de 2 éléments du formulaire. De **TABLEAU 1** découvrez les occurrences de 2-itemset.

Article	Compter
I1, I2	4
I1, I3	3
I1, I4	deux
I2, I3	4
I2, I4	3
I3, I4	deux

TABLEAU 2.4 Quatre. Étape de taille: montre que l'ensemble d'éléments {I1, I4} et {I3, I4} ne correspond pas à min_sup, il est donc supprimé..

Article	Compter
I1, I2	4
I1, I3	3
I2, I3	4
I2, I4	3

TABLEAU 2.5. Rejoignez et élaguez l'étape: Ensemble de 3 éléments du formulaire. Du **TABLEAU 1** trouver les occurrences de 3-itemset. De **TABLEAU 5** , découvrez les sous-ensembles de 2 éléments qui prennent en charge min_sup.

Nous pouvons voir que pour les sous-ensembles d'éléments {I1, I2, I3}, {I1, I2}, {I1, I3}, {I2, I3} se produisent dans **TABLEAU 5** ainsi {I1, I2, I3} est fréquent.

Nous pouvons voir que pour les sous-ensembles d'éléments {I1, I2, I4}, {I1, I2}, {I1, I4}, {I2, I4}, {I1, I4} ne sont pas fréquents, car ils ne se produisent pas dans **TABLEAU 5** ainsi {I1, I2, I4} n'est pas fréquent, donc il est supprimé.

Article
I1, I2, I3
I1, I2, I4
I1, I3, I4
I2, I3, I4

TABLEAU 2.6 Seul {I1, I2, I3} est fréquent .

À partir de l'ensemble d'éléments fréquemment découverts ci-dessus, l'association pourrait être:

{I1, I2} => {I3}

Confiance = support {I1, I2, I3} / support {I1, I2} = (3/4) * 100 = 75%

{I1, I3} => {I2}

Confiance = support {I1, I2, I3} / support {I1, I3} = (3/3) * 100 = 100%

{I2, I3} => {I1}

Confiance = support {I1, I2, I3} / support {I2, I3} = (3/4) * 100 = 75%

{I1} => {I2, I3}

Confiance = support {I1, I2, I3} / support {I1} = (3/4) * 100 = 75%

{I2} => {I1, I3}

Confiance = support {I1, I2, I3} / support {I2 = (3/5) * 100 = 60%

{I3} => {I1, I2}

Confiance = support {I1, I2, I3} / support {I3} = (3/4) * 100 = 75%

Cela montre que toutes les règles d'association ci-dessus sont fortes si le seuil de confiance minimum est de 60%.

Avantage

1- Algorithme facile à comprendre
2- Les étapes de jointure et d'élagage sont faciles à implémenter sur de grands ensembles d'éléments dans de grandes bases de données

2.2.5 Inconvénient

1- Cela nécessite un calcul élevé si les ensembles d'éléments sont très volumineux et que le support minimum est maintenu très bas.
2- La base de données entière doit être scannée. [8]

2.2.2. L'Algorithme Fp-growth

A été expliqué en détail dans notre tutoriel précédent. Dans ce tutoriel, nous allons en apprendre davantage sur la croissance fréquente des motifs – La croissance FP est une méthode d'extraction de jeux d'éléments fréquents.

Comme nous le savons tous, Apriori est un algorithme pour l'exploration fréquente de motifs qui se concentre sur la génération de jeux d'éléments et la découverte du jeu d'éléments le plus fréquent. Il réduit considérablement la taille de l'ensemble d'éléments dans la base de données, mais A priori a également ses propres lacunes.

1) La première étape consiste à analyser la base de données pour trouver les occurrences des ensembles d'éléments dans la base de données. Cette étape est la même que la première étape d'Apriori. Le nombre de 1-itemsets dans la base de données est appelé support count ou fréquence de 1-itemset.
2) La deuxième étape consiste à construire l'arbre FP. Pour cela, créez la racine de l'arbre. La racine est représentée par null.
3) L'étape suivante consiste à analyser à nouveau la base de données et à examiner les transactions. Examinez la première transaction et découvrez le jeu d'éléments qu'elle contient. L'ensemble d'éléments avec le nombre maximum est pris en haut, l'ensemble d'éléments suivant avec un nombre inférieur, etc. Cela signifie que la branche de l'arbre est construite avec des ensembles d'éléments de transaction dans l'ordre décroissant du nombre.
4) La transaction suivante dans la base de données est examinée. Les itemsets sont ordonnés par ordre décroissant de comptage. Si un ensemble d'éléments de cette transaction est déjà présent dans une autre branche (par exemple dans la 1ère transaction), alors cette branche de transaction partagerait un préfixe commun à la racine.

   Cela signifie que l'ensemble d'éléments commun est lié au nouveau nœud d'un autre ensemble d'éléments dans cette transaction.
5) De plus, le nombre d'items est incrémenté au fur et à mesure qu'il se produit dans les transactions. Le nombre de nœuds communs et de nouveaux nœuds est augmenté de 1 à mesure qu'ils sont créés et liés en fonction des transactions.
6) L'étape suivante consiste à extraire l'arborescence FP créée. Pour cela, le nœud le plus bas est examiné en premier avec les liens des nœuds les plus bas. Le noeud le plus bas représente la longueur du motif fréquentiel 1. À partir de là, parcourez le chemin dans

l'arborescence FP. Ce ou ces chemins sont appelés une base de motif conditionnelle. La base de modèle conditionnelle est une sous-base de données composée de chemins de préfixe dans l'arborescence FP se produisant avec le nœud le plus bas (suffixe).

7) Construire un arbre FP conditionnel, qui est formé par un nombre d'ensembles d'éléments dans le chemin. Les ensembles d'éléments répondant au seuil de prise en charge sont considérés dans l'arborescence FP conditionnelle.
8) Des modèles fréquents sont générés à partir de l'arbre FP conditionnel.

L'Algorithme FP-Growth: Pseudo Code

Procedure FP growth*(T)
Input: A conditional FP-tree T
Output: The complete set of all FT's corresponding to T.
Method:
1. **if** T only contains a single branch B
2. **for each** subset Y of the set of items in B
3. output itemset Y U T.base with count = smallest count of nodes in Y;
4. **else for each** i in T.header **do begin**
5. output Y = T.base U {i} with i.count;
6. **if** T.FP-array is defined
7. construct a new header table for Y's FP-tree from T.FP-array
8. **else** construct a new header table from T;
9. **construct** Y's conditional FP-tree Ty and possibly its FP-array Ay;
10. **if** Ty ≠ Ø
11. call FPgrowth *(Ty);
12. **end**

Exemple D'Algorithme De Croissance FP

Seuil de soutien=50%, Confiance= 60%

Transaction	Liste d'objets
T1	I1, I2, I3
T2	I2, I3, I4
T3	I4, I5
T4	I1, I2, I4
T5	I1, I2, I3, I5
T6	I1, I2, I3, I4

Tableau 2.7 Solution:
Seuil de support = 50% => 0,5
* 6 = 3 => min_sup = 3
1. Nombre de chaque

Article	Compter
I1	4
I2	5
I3	4
I4	4
I5	deux

Tableau 2.8 Triez l'ensemble d'éléments par ordre décroissant.article.

Article	Compter
I2	5
I1	4
I3	4
I4	4

Tableau 2.9 Construire l'arbre FP

3. Construire un Arbre FP

1. Considérant le nœud racine nul.
2. La première analyse de la transaction T1: I1, I2, I3 contient trois éléments {I1:1}, {I2:1}, {I3: 1}, où I2 est lié en tant qu'enfant à root, I1 est lié à I2 et I3 est lié à I1.
3. T2: I2, I3, I4 contient I2, I3 et I4, où I2 est lié à root, I3 est lié à I2 et I4 est lié à I3. Mais cette branche partagerait le nœud I2 aussi commun qu'il est déjà utilisé dans T1.
4. Incrémentez le nombre de I2 de 1 et I3 est lié en tant qu'enfant à I2, I4 est lié en tant qu'enfant à I3. Le compte est {I2:2}, {I3: 1}, {I4: 1}.
5. T3: I4, I5. De même, une nouvelle branche avec I5 est liée à I4 lors de la création d'un enfant.
6. T4: I1, I2, I4. La séquence sera I2, I1 et I4. I2 est déjà lié au nœud racine, il sera donc incrémenté de 1. De même I1 sera incrémenté de 1 car il est déjà lié à I2 dans T1, donc {I2:3}, {I1:2}, {I4: 1}.
7. T5: I1, I2, I3, I5. La sequence sera I2, I1, I3 et I5. Ainsi {I2:4}, {I1: 3}, {I3: 2}, {I5: 1}.
8. T6: I1, I2, I3, I4. La sequence sera I2, I1, I3 et I4. Anisi {I2:5}, {I1: 4}, {I3: 3}, {I4 1}.

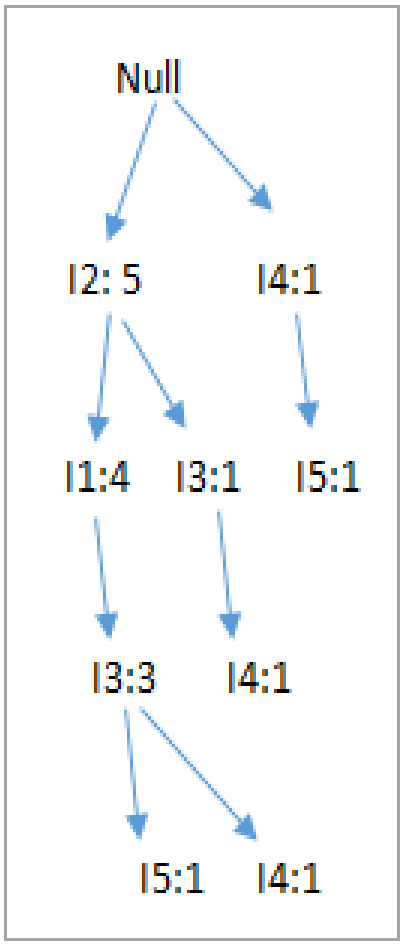

Figure2.1 FP-Arbre

4. L'extraction de l'arbre FP est résumée ci-dessous:

1. L'élément de noeud le plus bas I5 n'est pas considéré car il n'a pas de compte de support minimum, il est donc supprimé.
2. Le nœud inférieur suivant est I4. I4 se produit dans 2 branches, {I2, I1, I3:, I41}, {I2, I3, I4: 1}. Par conséquent, en considérant I4 comme suffixe, les chemins de préfixe seront {I2, I1, I3:1}, {I2, I3: 1}. Cela forme la base du modèle conditionnel.
3. La base de modèle conditionnelle est considérée comme une base de données de transactions, un arbre FP est construit. Cela contiendra {I2: 2, I3:2}, I1 n'est pas considéré car il ne répond pas au nombre de support minimum.
4. Ce chemin générera toutes les combinaisons de motifs fréquents: {I2, I4: 2}, {I3, I4: 2}, {I2, I3, I4: 2}

5. Pour I3, le chemin de préfixe serait: {I2, I1: 3}, {I2: 1}, cela générera un arbre FP à 2 nœuds: {I2:4, I1: 3} et des motifs fréquents sont générés: {I2,I3:4}, {I1:I3:3}, {I2,I1,I3: 3}.
6. Pour I 1, le chemin de préfixe serait: {I 2:4} cela générera un seul nœud FP-tree: {I 2:4} et des motifs fréquents sont générés: {I2, I1: 4}.

Article	Base de modèle conditionnelle	Arbre FP conditionnel	Modèles fréquents générés
I4	{I2, I1, I3: 1}, {I2, I3: 1}	{I2: 2, I3: 2}	{I2, I4: 2}, {I3, I4: 2}, {I2, I3, I4: 2}
I3	{I2, I1: 3}, {I2: 1}	{I2: 4, I1: 3}	{I2, I3: 4}, {I1: I3: 3}, {I2, I1, I3: 3}
I1	{I2: 4}	{I2: 4}	{I2, I1: 4}

TABLEAU 2.10 Le diagramme donné ci-dessous représente l'arbre FP conditionnel associé au nœud conditionnel I3.

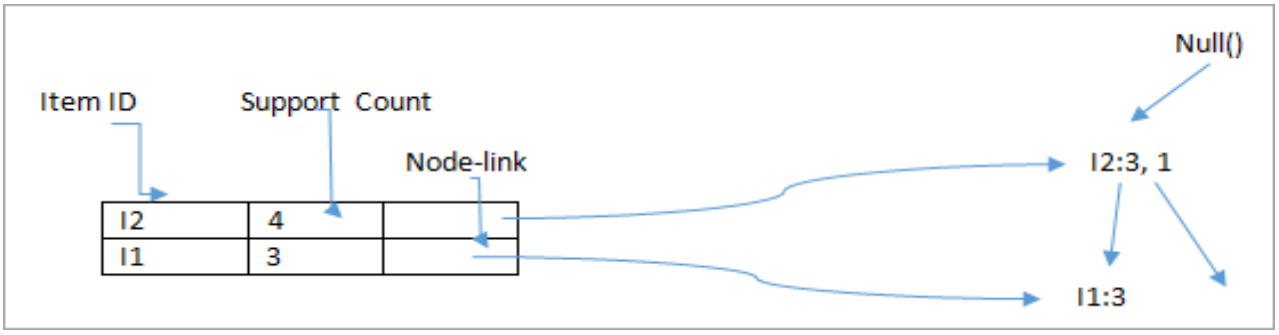

Figure2.2 arbre-FP-conditionnel-associé-au-nœud-conditionnel-I3

Avantages

Cet algorithme ne doit analyser la base de données que deux fois par rapport à Apriori qui analyse les transactions pour chaque itération.

1) L'appariement des éléments ne se fait pas dans cet algorithme et cela le rend plus rapide.
2) La base de données est stockée dans une version compacte en mémoire.
3) Il est efficace et évolutif pour l'extraction de modèles fréquents longs et courts.

Inconvénient

1) L'arbre FP est plus lourd et difficile à construire qu'Apriori.
2) Cela peut coûter cher.
3) Lorsque la base de données est volumineuse, l'algorithme peut ne pas tenir dans la mémoire partagée.[9]

2.2.3. L'Algorithme Eclat

Eclat est un algorithme permettant de découvrir des ensembles d'éléments fréquents dans une base de données de transactions. Il a été proposé par Zaki (2001). Contrairement à des algorithmes tels qu'Apriori, Eclat utilise une recherche en profondeur d'abord pour découvrir des ensembles d'éléments fréquents au lieu d'une recherche en profondeur d'abord.

L'Algorithme Eclat : Pseudo Code

input: alphabet A with ordering $\leq$,
multiset T $\subseteq$ P(A) of sets of items,
minimum support value minsup $\in \mathbb{N}$;
output: set F of frequent itemsets and their support counts.
1) For (i=1, i $\leq$ data. Length) do
2) F:= {(Ø,|T|};
3) C_0:= {(x, T({x}))| x $\in$ A};
4) $C^i_\emptyset$:= freq ($c_\emptyset$):= {(x, Tx) (x, Tx) $\in C_0$,|Tx| <minsup};
5) F:= { Ø };
6) addFrequentSupersets (Ø,C^i_q) ;
7) end for
function addFrequentSupersets:
Input: frequent itemset p $\in$ P(A) called prefix,
Incidence matrix C of frequent 1-item-extensions ofp.
Output: add all frequent extensions of p to global variable F.
8) **for** (x, Tx) $\in$ C **do**
9) q:=p U{x};
10) c_q= {(y, Tx $\cap$ Ty) | (y, Ty) $\in$ C, y>x } ;
11) C^i_q= freq(Cq):= {(y, Ty) | (y. Ty) $\in$ Cq.| Ty |$\geq$ yminsup};
12) **if** C^i_q =Øthen
3) addFrequentSupersets (q, C^i_q);
14) **end if**
15) F:=F[{(q. | Tx.|)};
16) **end for**

L'entrée de l'algorithme Eclat

L'entrée est une base de données de transactions (alias contexte binaire) et un seuil nommé minsup (une valeur comprise entre 0 et 100%).

Une base de données de transactions est un ensemble de transactions. Chaque transaction est un ensemble d'éléments. Par exemple, considérons la base de données de transactions suivante. Il contient 5 transactions (t1, t2,..., t5) et 5 articles (1,2, 3, 4, 5). Par exemple, la première transaction représente l'ensemble des items 1, 3 et 4. Cette base de données est fournie sous la forme du fichier contextPasquier99.txt dans la distribution SPMF. Il est important de noter qu'un élément n'est pas autorisé à apparaître deux fois dans la même transaction et que les éléments sont supposés être triés par ordre lexicographique dans une transaction.

Id de transaction	Article
t1	{1, 3, 4}
t2	{2, 3, 5}
t3	{1, 2, 3, 5}
t4	{2, 5}
t5	{1, 2, 3, 5}

TABLEAU 2.11 L'entrée de l'algorithme Eclat

la sortie de l'algorithme Eclat

Eclat est un algorithme permettant de découvrir des itemsets (groupes d'items) se produisant fréquemment dans une base de données de transactions (frequent itemsets). Un ensemble d'éléments fréquent est un ensemble d'éléments apparaissant dans au moins les transactions minsup de la base de données de transactions, où minsup est un paramètre donné par l'utilisateur.

Par exemple, si Eclat est exécuté sur la base de données de transactions précédente avec un minsup de 40 % (2 transactions), Eclat produit le résultat suivant:

ensembles d'items	support
{1}	3
{2}	4
{3}	4
{5}	4
{1, 2}	2
{1, 3}	3
{1, 5}	2
{2, 3}	3
{2, 5}	4
{3, 5}	3
{1, 2, 3}	2
{1, 2, 5}	2
{1, 3, 5}	2
{2, 3, 5}	3
{1, 2, 3, 5}	2

TABLEAU 2.12 la sortie de l'algorithme Eclat

Interpréter les résultats

Chaque item fréquent est annoté avec son support. La prise en charge d'un ensemble d'éléments est le nombre de fois que l'ensemble d'éléments apparaît dans la base de données de transactions. Par exemple, le jeu d'items {2, 3 5} a un support de 3 car il apparaît dans les transactions t2, t3 et t5. C'est un ensemble d'éléments fréquent car sa prise en charge est supérieure ou égale au paramètre minsup.

Format de fichier d'entrée

Le format de fichier d'entrée utilisé par ECLAT est défini comme suit. C'est un fichier texte. Un élément est représenté par un entier positif. Une transaction est une ligne du fichier texte. Dans chaque ligne (transaction), les éléments sont séparés par un seul espace. On suppose que tous les éléments d'une même transaction (ligne) sont triés selon un ordre total (par exemple, ordre croissant) et qu'aucun élément ne peut apparaître deux fois dans la même ligne.

Par exemple, pour l'exemple précédent, le fichier d'entrée est défini comme suit:

```
1 3 4
2 3 5
1 2 3 5
2 5
1 2 3 5
```

Notez qu'il est également possible d'utiliser le format ARFF comme alternative au format d'entrée par défaut. La spécification du format ARFF peut être trouvée ici. La plupart des fonctionnalités du format ARFF sont prises en charge sauf que (1) le caractère "=" est interdit et (2) les caractères d'échappement ne sont pas pris en compte. Notez que lorsque le format ARFF est utilisé, les performances des algorithmes d'exploration de données seront légèrement inférieures à celles du format de fichier SPMF natif, car une conversion du fichier d'entrée sera automatiquement effectuée avant de lancer l'algorithme et le résultat devra également être converti. Ce coût devrait cependant être faible.

Format de fichier de sortie

Le format de fichier de sortie est défini comme suit. Il s'agit d'un fichier texte, où chaque ligne représente un ensemble d'éléments fréquent. Sur chaque ligne, les éléments de l'ensemble d'éléments sont d'abord répertoriés. Chaque élément est représenté par un entier et il est suivi d'un seul espace. Après, tous les items, le mot clé "#SUP:" apparaît, qui est suivi d'un entier indiquant la prise en charge du jeu d'items, exprimé en nombre de transactions. Par exemple, voici le fichier de sortie pour cet exemple. La première ligne indique le jeu d'éléments fréquent constitué de l'élément 1 et indique que ce jeu d'éléments prend en charge 3 transactions.

```
1 # SUP: 3
2 # SUP: 4
3 # SUP: 4
5 # SUP: 4
1 2 # SUP:2
1 3 # SUP: 3
1 5 # SUP: 2
2 3 # SUP: 3
2 5 # SUP: 4
3 5 # SUP: 3
1 2 3 #SUP: 2
1 2 5 #SUP: 2
1 3 5 #SUP: 2
2 3 5 #SUP: 3
1 2 3 5 #SUP: 2
```

Notez que si le format ARFF est utilisé en entrée au lieu du format d'entrée par défaut, le format de sortie sera le même sauf que les éléments seront représentés par des chaînes au lieu d'entiers. [10]

2.2.4.l'Algorithme INDIRECT

Indirecte est un algorithme permettant de découvrir des associations indirectes entre des éléments dans des bases de données de transactions.

Pourquoi cet algorithme est-il important? Parce que les algorithmes traditionnels d'exploration de règles d'association se concentrent sur les associations directes entre les ensembles d'éléments. Cet algorithme peut découvrir des associations indirectes, qui peuvent être utiles dans des domaines tels que la biologie. L'exploitation minière par association indirecte a diverses applications telles que l'analyse des marchés boursiers et l'analyse des produits concurrentiels. [11]

L'Algorithme Indirecte: Pseudo Code

Extract frequent itemsets ,L_1 ,L_2,….L_n ,using frequent itemset generation algorithm, where L_i is the set of all frequent i-itemsets.
2. P = Ø (set of indirect associations)
3. for k=2to n do
4. c_{k+1}= join (L_k , L_k)
5. for each< x, y, M > ∈ c_{k+1} do
6. if sup({x, y}) <T_S and dep({x}, M) ≥ T_d and dep({y}, M) ≥ T_d)
7. P = P ∪{< x, y ,M >}
8. End
9. end
10. end

Entrée :

L'entrée de l'algorithme indirect est une base de données de transactions et trois paramètres nommés minsup (une valeur dans [0,1] qui représente un pourcentage), ts (une valeur dans [0,1] qui représente un pourcentage) et minconf (une valeur dans [0,1] qui représente un pourcentage).

Une base de données de transactions est un ensemble de transactions. Chaque transaction est un ensemble d'éléments distincts. Par exemple, considérons la base de données de transactions suivante. Il contient 5 transactions (t1, t2,..., t5) et 5 articles (1,2, 3, 4, 5). Par example, la première transaction représente l'ensemble des items 1, 4 et 5. Cette base de données est fournie sous forme de fichier contextIndirect.txt dans la distribution SPMF. Il est important de noter qu'un élément n'est pas autorisé à apparaître deux fois dans la même transaction et que les éléments sont supposés être triés par ordre lexicographique dans une transaction. [11]

Id de transaction	**Article**
t1	{1, 4,5}
t2	{2, 3, 4}
t3	{1, 2, 4, 5}
t4	{5}
t5	{1, 2, 4, 5}

TABLEAU 2.12 Entrée INDIRECT

Les trois paramètres numériques de l'algorithme indirect sont:

- minsup: appelé le "support minimum du médiateur".
- ts: appelé le "support minimum de la paire d'éléments"
- minconf: représentant la confiance minimale requise pour les associations indirectes (notez que dans l'article original, il utilise la mesure IS au lieu de la confiance). [11]

Sortie :

Le résultat est toutes les associations indirectes respectant les paramètres minsup, ts et minconf. Une association indirecte a la forme {x, y} ==> M, où x et y sont des éléments uniques et M est un ensemble d'éléments appelé "médiateur".

Une association indirecte doit respecter les conditions suivantes:

- Le nombre de transactions contenant tous les éléments de {x} M M divisé par le nombre total de transactions doit être supérieur ou égal à minsup.
- Le nombre de transactions contenant tous les éléments de {y} M M divisé par le nombre total de transactions doit être supérieur ou égal à minsup.
- Le nombre de transactions contenant {x, y} divisé par le nombre total de transactions doit être inférieur à ts.
- La confiance de {x}par rapport à M et {y}par rapport à M doit être supérieure ou égale à minconf. La confiance d'un ensemble d'éléments X par rapport à un autre ensemble d'éléments Y est définie comme le nombre de transactions contenant X et Y

Divisé par le nombre de transactions contenant X.

Par exemple, en appliquant l'algorithme indirect avec minsup = 60 %, ts = 50 % et minconf= 10%, nous obtenons 3 règles d'association indirectes:

1. {1, 2 | {4}}, ce qui signifie que 1 et 2 sont indirectement associés par le médiateur {4 }.
2. {1, 5 | {4}}, ce qui signifie que 1 et 5 sont indirectement associés par le médiateur {4 }.
3. {2, 5 | {4}}, ce qui signifie que 1 et 5 sont indirectement associés par le médiateur {4 }.

Pour plus de détails sur chacune de ces trois règles indirectes, exécutez cet exemple.

Format de fichier d'entrée

Le format de fichier d'entrée est un fichier texte contenant des transactions. Chaque ligne représente une transaction. Les éléments de la transaction sont répertoriés sur la ligne correspondante. Un élément est représenté par un entier positif. Chaque élément est séparé de l'élément suivant par un espace. On suppose que les articles sont triés en fonction d'une commande totale et qu'aucun article ne peut apparaître deux fois dans la même transaction. Par exemple, pour l'exemple précédent, le fichier d'entrée est défini comme suit :

1 4
2 3 4
1 2 4 5
4 5
1 2 4 5

Ce fichier contient cinq lignes (cinq transactions). Considérez la première ligne. Cela signifie que la première transaction est l'ensemble d'éléments {1, 4}. Les lignes suivantes suivent le même format.

Notez qu'il est également possible d'utiliser le format ARFF comme alternative au format d'entrée par défaut. La spécification du format ARFF peut être trouvée ici. La plupart des fonctionnalités du format ARFF sont prises en charge sauf que (1) le caractère "=" est interdit et (2) les caractères d'échappement ne sont pas pris en compte. Notez que lorsque le format ARFF est utilisé, les performances des algorithmes d'exploration de données seront

légèrement inférieures à celles du format de fichier SPMF natif, car une conversion du fichier d'entrée sera automatiquement effectuée avant de lancer l'algorithme et le résultat devra également être converti. Ce coût devrait cependant être faible. [11]

Format de fichier de sortie

Le format de fichier de sortie est défini comme suit. Il s'agit d'un fichier texte, où chaque ligne représente une règle d'association indirecte. Chaque ligne commence par "(a=x b=y | m=M) " indiquant que la ligne représente la règle {x,y} ==> M, où x, y et M sont des entiers représentant des éléments. Ensuite, le mot clé "#sup(a,a)=" est suivi du support de {x} M M exprimé en nombre de transactions (un entier). Ensuite, le mot clé "#sup(b,b)=" est suivi du support de {y} M M exprimé en nombre de transactions (un entier). Ensuite, le mot clé "#conf(a,mediator)= " est suivi de la confiance de a par rapport au médiateur, exprimée sous la forme d'une valeur double dans l'intervalle [0, 1]. Ensuite,le mot clé "#conf(b, mediator)= " apparaît suivi de la confiance de b par rapport au médiateur, exprimée sous la forme d'une valeur double dans l'intervalle [0, 1].

Par exemple, le fichier de sortie de cet exemple est:

(a=1 b=2 / médiateur=4) #sup (a, médiateur)= 3 #sup(b, médiateur)= 3 #conf (a, médiateur)= 1.0 #conf (b, médiateur)= 1.0

(a=1 b=5 / médiateur=4) #sup (a, médiateur)= 3 #sup(b, médiateur)= 3 #conf (a, médiateur)= 1.0 #conf (b, médiateur)= 1.0

(a=2 b=5 / médiateur=4) #sup (a, médiateur)= 3 #sup(b, médiateur)= 3 #conf (a, médiateur)= 1.0 #conf (b, médiateur)= 1.0

Ce fichier contient trois lignes (trois règles d'association indirecte). Considérez la première ligne. Il représente que les articles 1 et 2 sont indirectement associés par l'article 4 en tant que médiateur. De plus, il indique que le support de {1, 4} est de 3 transactions, le support de {2,4} est de 3 transactions, la confiance de l'article 1 par rapport à l'article 4 est de 100% et la confiance de l'article 2 par rapport à l'article 4 est de 100%. Les autres lignes suivent le même format.

Notez que si le format ARFF est utilisé en entrée au lieu du format d'entrée par défaut, le format de sortie sera le même sauf que les éléments seront représentés par des chaînes au lieu d'entiers. [11]

2.2.4. l'Algorithme H-MINe :

H-Mine est un algorithme de découverte d'éléments fréquents dans les bases de données de transactions, proposé par Pei et al. (2001). Contrairement aux algorithmes précédents tels qu'Apriori, H-Mine utilise une approche de croissance de modèle pour découvrir des ensembles d'éléments fréquents.[12]

L'Algorithme H-Mine: Pseudo Code

Step 1.
Scan the transaction database TDB once to find L, the complete set of frequent items.
Step 2. P
Partition TDB into k parts,TDB_1, ...,TDB_k, such that, for each,TDB_i ($1 \leq i \leq k$), the frequent-item projections in TDB_ican be held in the main memory
Step 3
For i = 1 to k, use H-mine(Mem) to mine frequent patterns in $TDB_i$with respect to the minimum support threshold min sup =[min sup × n_i/n],
Where n and $n_i$are the number of transactions in TDB and
Step 4.
TDB_i, respectively.Let F_i be the set of frequent patterns inTDB_i.
Let F = $U_{i=1}^{k}$ F_i. Scan TDB one more time, collect support for patterns in F.
Output those patterns which pass the minimum support threshold min sup.

Entrée

L'entrée de H-Mine est une base de données de transactions (alias contexte binaire) et un seuil nommé minsup (une valeur comprise entre 0 et 100%).

Une base de données de transactions est un ensemble de transactions. Chaque transaction est un ensemble d'éléments. Par exemple, considérons la base de données de transactions suivante. Il contient 5 transactions (t1, t2,..., t5) et 5 articles (1,2, 3, 4, 5). Par example, la première transaction représente l'ensemble des items 1, 3 et 4. Cette base de données est fournie sous la forme du fichier contextPasquier99.txt dans la distribution SPMF. Il est

important de noter qu'un élément n'est pas autorisé à apparaître deux fois dans la même transaction et que les éléments sont supposés être triés par ordre lexicographique dans une transaction.

Transaction id	Items
t1	{1, 3, 4}
t2	{2, 3, 5}
t3	{1, 2, 3, 5}
t4	{2, 5}
t5	{1, 2, 3, 5}

TABLEAU 2.13 Entrée H-mine

Sortie

H-Mine est un algorithme permettant de découvrir des itemsets (groupes d'items) se produisant fréquemment dans une base de données de transactions (frequent itemsets). Un ensemble d'éléments fréquent est un ensemble d'éléments apparaissant dans au moins les transactions minsup de la base de données de transactions, où minsup est un paramètre donné par l'utilisateur.

Par exemple, si H-Mine est exécuté sur la base de données de transactions précédente avec un minsup de 40 % (2 transactions), H-Mine produit le résultat suivant:

itemsets	support
{1}	3
{2}	4
{3}	4
{5}	4
{1, 2}	2
{1, 3}	3
{1, 5}	2
{2, 3}	3
{2, 5}	4
{3, 5}	3
{1, 2, 3}	2
{1, 2, 5}	2
{1, 3, 5}	2
{2, 3, 5}	3
{1, 2, 3, 5}	2

BLEAU 2.12 Sortie H-mine

Format de fichier d'entrée

Le format de fichier d'entrée utilisé par H-Mine est défini comme suit. C'est un fichier texte. Un élément est représenté par un entier positif. Une transaction est une ligne du fichier texte. Dans chaque ligne (transaction), les éléments sont séparés par un seul espace. On suppose que tous les éléments d'une même transaction (ligne) sont triés selon un ordre total (par

exemple, ordre croissant) et qu'aucun élément ne peut apparaître deux fois dans la même ligne.

Par exemple, pour l'exemple précédent, le fichier d'entrée est défini comme suit:

```
1 3 4
2 3 5
1 2 3 5
2 5
1 2 3 5
```

Notez qu'il est également possible d'utiliser le format ARFF comme alternative au format d'entrée par défaut. La spécification du format ARFF peut être trouvée ici. La plupart des fonctionnalités du format ARFF sont prises en charge sauf que (1) le caractère "=" est interdit et (2) les caractères d'échappement ne sont pas pris en compte. Notez que lorsque le format ARFF est utilisé, les performances des algorithmes d'exploration de données seront légèrement inférieures à celles du format de fichier SPMF natif, car une conversion du fichier d'entrée sera automatiquement effectuée avant de lancer l'algorithme et le résultat devra également être converti. Ce coût devrait cependant être faible.

Format de fichier de sortie

Le format de fichier de sortie est défini comme suit. Il s'agit d'un fichier texte, où chaque ligne représente un ensemble d'éléments fréquent. Sur chaque ligne, les éléments de l'ensemble d'éléments sont d'abord répertoriés. Chaque élément est représenté par un entier et il est suivi d'un seul espace. Après, tous les items, le mot clé "#SUP:" apparaît, qui est suivi d'un entier indiquant la prise en charge du jeu d'items, exprimé en nombre de transactions. Par exemple, voici le fichier de sortie pour cet exemple. La première ligne indique le jeu d'éléments fréquent constitué de l'élément 1 et indique que ce jeu d'éléments prend en charge 3 transactions.

```
1 #SUP: 3
2 #SUP: 4
3 #SUP: 4
5 #SUP: 4
1 2 #SUP: 2
1 3 #SUP: 3
1 5 #SUP: 2
2 3 #SUP: 3
2 5 #SUP: 4
3 5 #SUP: 3
1 2 3 #SUP: 2
1 2 5 #SUP: 2
1 3 5 #SUP: 2
2 3 5 #SUP: 3
1 2 3 5 #SUP: 2
```

Notez que si le format ARFF est utilisé en entrée au lieu du format d'entrée par défaut, le format de sortie sera le même sauf que les éléments seront représentés par des chaînes au lieu d'entiers.

CHAPITRE 03 : IMPLÉMENTATION

3.2 **Introduction** :

Dans ce chapitre après l'aperçu théorique général des chapitres précédents, nous vous présentons Les technologies et Les outils sont utilisés Dans la partie pratique du mémoire

3.3 Les technologies :

3.2.1 Open sourcing (git + github)

3.2.1.1 **Git**

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. Le principal contributeur actuel de git et depuis plus de 16 ans est Junio C Hamano. En 2016, il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes.[13]

3.2.1.2 **Github**

GitHub, Inc. est un fournisseur d'hébergement Internet pour le développement de logiciels et le contrôle de version utilisant Git. Il offre les fonctionnalités de contrôle de version distribué et de gestion du code source (SCM) de Git, ainsi que ses propres fonctionnalités. Il fournit un contrôle d'accès et plusieurs fonctionnalités de collaboration telles que le suivi des bogues, les demandes de fonctionnalités, la gestion des tâches, l'intégration continue et les wikis pour chaque projet. Basée en Californie, elle est une filiale de Microsoft depuis 2018.

Il est couramment utilisé pour héberger des projets open source. En novembre 2021, GitHub comptait plus de 73 millions de développeurs et plus de 200 millions de référentiels (dont au moins 28 millions de référentiels publics). C'est le plus grand hôte de code source en novembre 2021. [14]

3.2.2 **PHP** :

Le PHP, pour Hypertext Preprocessor, désigne un langage informatique, ou un langage de script, utilisé principalement pour la conception de sites web dynamiques. Il s'agit d'un langage de programmation sous licence libre qui peut donc être utilisé par n'importe qui de façon totalement gratuite.

Créé au début des années 1990 par le Canadien et Groenlandais Rasmus Lerdorf, le langage PHP est souvent associé au serveur de base de données MySQL et au serveur

Apache. Avec le système d'exploitation Linux, il fait partie intégrante de la suite de logiciels libres LAMP.

Sur un plan technique, le PHP s'utilise la plupart du temps côté serveur. Il génère du code HTML, CSS ou encore XHTML, des données (en PNG, JPG, etc.) ou encore des fichiers PDF. Il fait, depuis de nombreuses années, l'objet d'un développement spécifique et jouit aujourd'hui une bonne réputation en matière de fiabilité et de performances. [15]

3.2.2.2 Avantages de PHP

- PHP a certains avantages qui l'ont rendu si populaire, et c'est le langage de prédilection pour les serveurs Web depuis plus de 15 ans maintenant. Voici quelques-uns des avantages de PHP:
- Multiplateforme: PHP est indépendant de la plate-forme. Vous n'avez pas besoin d'avoir un système d'exploitation particulier pour l'utiliser, car il fonctionne sur toutes les plates-formes, qu'il s'agisse de Mac, Windows ou Linux.
- Open Source: PHP est open source. Le code original est mis à la disposition de tous ceux qui souhaitent s'en inspirer. C'est l'une des raisons pour lesquelles l'un de ses frameworks, Laravel, est si populaire.
- Facile à apprendre: PHP n'est pas difficile à apprendre pour les débutants absolus. Vous pouvez le ramasser assez si vous avez déjà des connaissances en programmation.
- PHP se synchronise avec toutes les bases de données: Vous pouvez facilement connecter PHP à toutes les bases de données, relationnelles et non relationnelles. Il peut donc se connecter en un rien de temps à MySQL, Postgress, MongoDB ou toute autre base de données.
- Communauté de soutien: PHP a une communauté en ligne très favorable. La documentation officielle fournit des guides sur la façon d'utiliser les fonctionnalités et vous pouvez facilement résoudre votre problème lorsque vous êtes bloqué.[15]

3.3 Les diagrammes de classes Les algorithmes :

3.3.1 L'algorithme Apriori

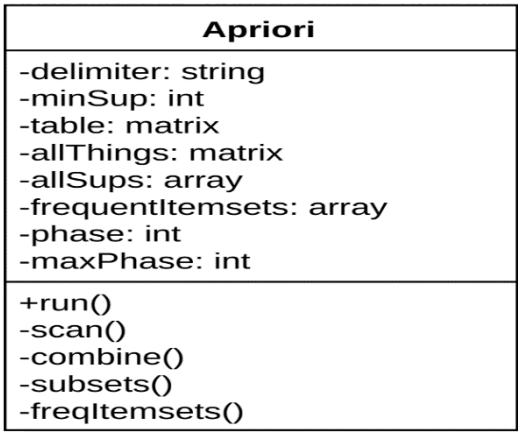

Figure3.1 Les diagramme de classe L'algorithme apriori

3.3.1 L'algorithme Eclat

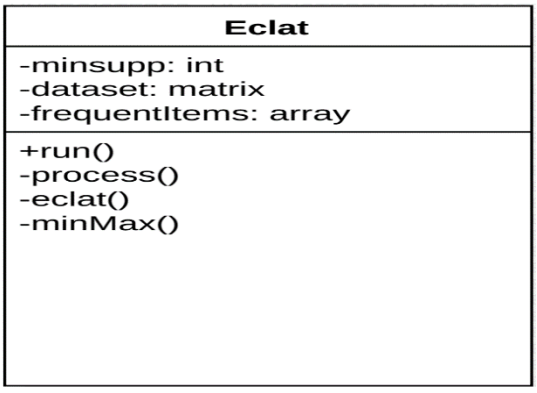

Figure3.2 Les diagramme de classe L'algorithme Eclat

3.3.1 L'algorithme FPGowth

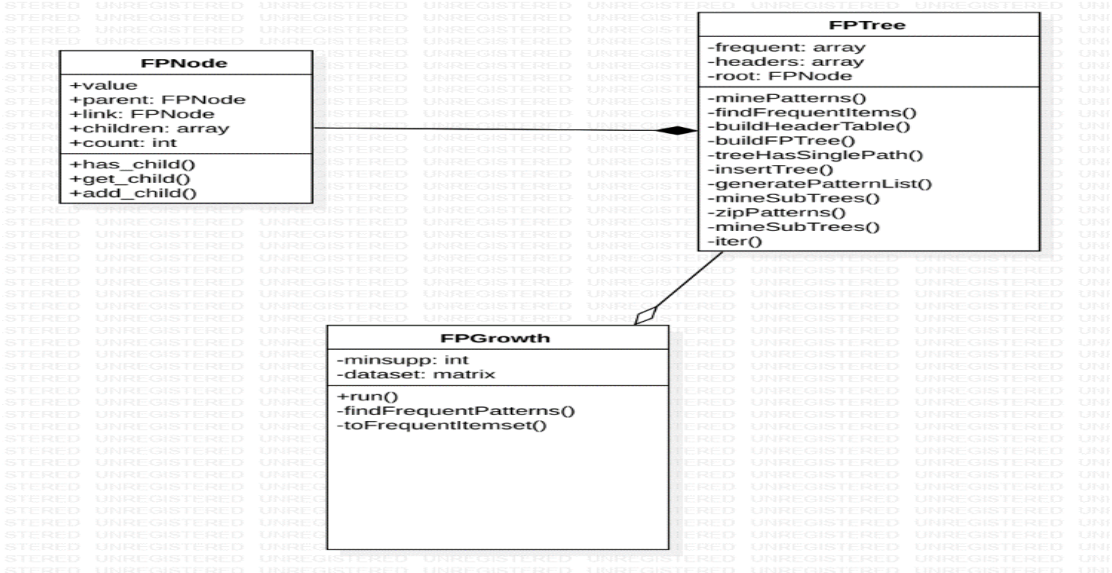

Figure3.3 Les diagramme de classe L'algorithme FPGowth

3.3.1 L'algorithme Indrect

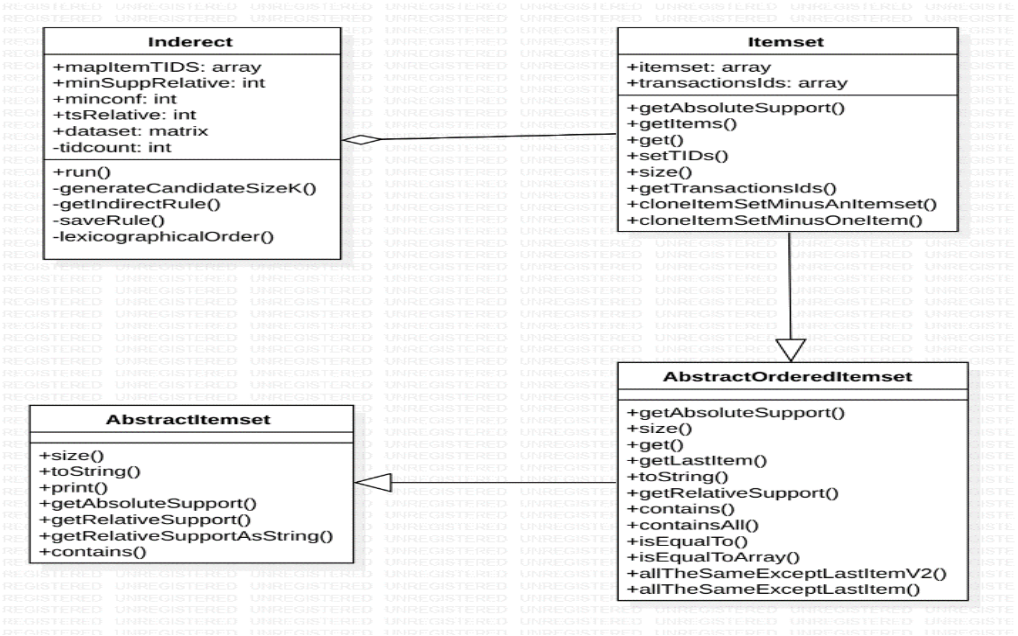

Figure3.4 Les diagramme de classe L'algorithme Indrect

3.3.1 L'algorithme HMine

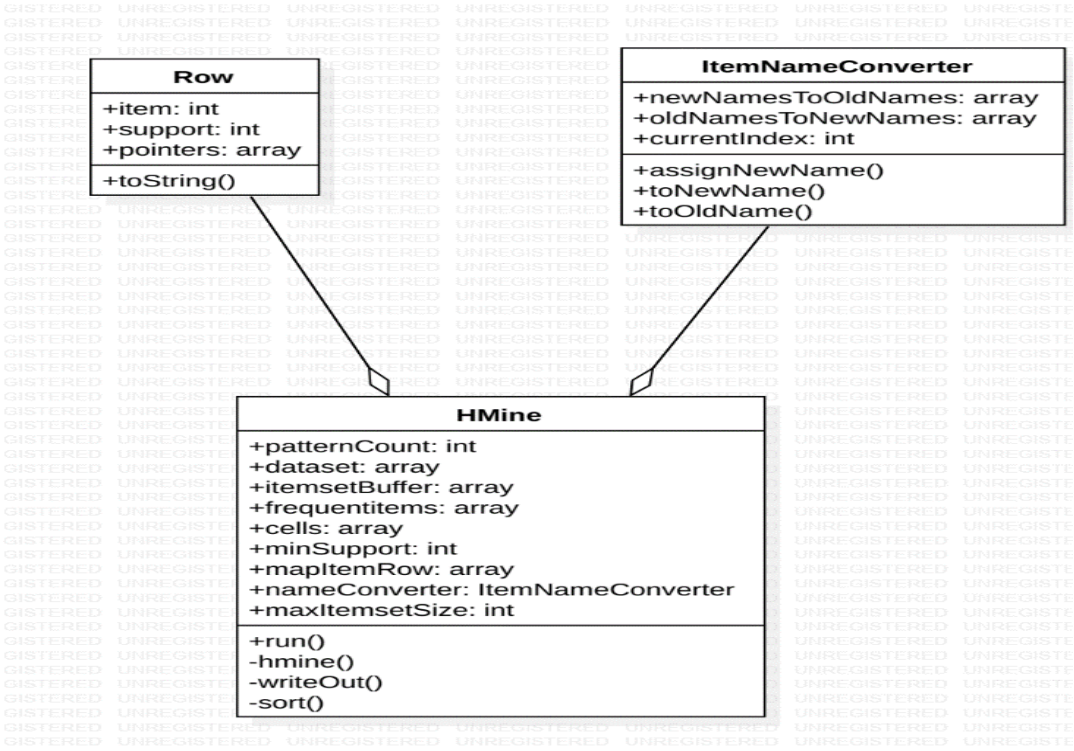

Figure3.5Les diagramme de classe L'algorithme HMine

3.4 L'Implémentation en PHP :

nous vous présentons notre travail dans ce dépôt de github

https://github.com/ziadmahdi/association-rules-mining

3.5 Conclusion :

Dans ce chapitre, nous avons vu les différents outils utilisés pour mettre en œuvre et Les diagrammes de classes.

CHAPITRE 03 : COMPARAISON ET RÉSULTAT

Chapitre 04 :Résultat et comparaison

4.1 Introduction

Dans ce dernier chapitre après l'aperçu théorique général des chapitres précédents, nous offrons un côté pratique pour appliquer notre objectif. Il est un système de de comparaison entre Les algorithmes exactes dans langage JAVA(SPMF) et PHP en termes de mieux la vitesse de mise en œuvre et consommation moins de mémoire

4.3 Spmf :

SPMF est un logiciel open source et une bibliothèque de data mining écrite en Java, spécialisée dans le pattern mining (la découverte de patterns dans les données) Il est distribué sous la licence GPLv3.

Il propose des implémentations de 234 algorithmes d'exploration de données pour:

- Exploitation minière de règles d'association,
- Extraction d'ensembles d'articles,
- Motif séquentiel
- Extraction de règles séquentielles,
- Prédiction de séquence,
- Extraction de motifs périodiques,
- Extraction d'épisodes
- extraction de motifs à haute utilité,
- extraction de séries chronologiques.
- clustering et classification,

SPMF peut être utilisé comme un programme autonome avec une interface utilisateur simple ou à partir de la ligne de commande.

De plus, le code source de chaque algorithme peut être facilement intégré dans d'autres logiciels Java.

En outre, certains wrappers non officiels sont disponibles pour d'autres langages tels que Python, R et Weka.

SPMF est rapide et léger (aucune dépendance à d'autres bibliothèques).

La version actuelle est la v2.54 et a été publiée le 14 juin 2022.[16]

4.4Les data sets :

Data sets	les transactions	items
basketball	96	25
BF	252	130
QK	2178	40

Les caractéristiques de la machine utilisée pour les tests :

CPU : i3 9100f

Ram : 16 Gb 2666 mh

Disk : ssd

4.5 Résultat et comparaison :

Il représentera les résultats obtenus à l'ordre du jour des éclaircissements plus

Min_sup = 01 ; min_conf =04 ; ts =0.1 ;

algorithme	Temps(ms)		Mémoire(mb)	
	PHP	JAVA	PHP	JAVA
apriori	8.8	1	0.309	0.186
Fp-growth	2	2	0.353	0.192
Eclat	1.5	1	0.177	0.214
INDIRECT	1.5	1	0.195	/
H-mine	2.4	3	0.108	0.077

- TABLEAU 4.1 : Résultat obtenus après l'exécution sur la base des données basketball

Dans ce test, nous dérivons des résultats du langage de programmation JAVA
Bon par rapport PHP

algorithme	Temps(ms)		Mémoire(mb)	
	PHP	JAVA	PHP	JAVA
apriori	25	6	3	0.160
Fp-growth	8	4	3	0.180
Eclat	26	4	3	0.192
INDIRECT	60	7	2.7	/
H-mine	241	7	2.9	0.200

TABLEAU 4.2 : Résultat obtenus après l'exécution sur la base des données QK

Dans ce test, nous dérivons des résultats du langage de programmation JAVA
Bon par rapport PHP

algorithme	Temps(ms)		Mémoire(mb)	
	PHP	JAVA	PHP	JAVA
apriori	300	10	2.1	0.108
Fp-growth	19	8	3	0.144
Eclat	55	7	1.3	0.120
INDIRECT	77	24	1	-
H-mine	19	7	0.9	0.138

TABLEAU 4.3 : Résultat obtenus après l'exécution sur la base des données BF

Dans ce test, nous dérivons des résultats du langage de programmation JAVA
Bon par rapport PHP

4.6 Comparaison

En basant sur le résultat obtenus précédent, on peut être établir une comparaison entre les deux langage de programmation avec les mêmes paramètres et différents L'Algorithmes et base des données.

- ✓ PHP coûteux en terme de temps d'exécution est espace mémoire par rapport à langage JAVA.
- ✓ java Très flexible par rapport à PHP.

4.7 Conclusion

Dans ce chapitre, nous avons vu les résultats de la mise en œuvre des algorithmes précédents dans le langage de programmation PHP et après comparaison entre les résultats, nous avons conclu que le langage de programmation a un impact significatif sur l'efficacité et la qualité du projet en termes de temps d'exécution et de consommation de mémoire.

Conclusion générale

Dans le cadre de ce travail de fin d'étude, nous avons découvert des domaines complètement nouveaux telle la fouille de données et le monde de la recherche. Cette expérience nous a amenés à comprendre les diverses problématiques liées aux de ces algorithmes en fonction de différents langages JAVA et PHP.

Lors de ce travail, nous avons essayé d'étudier et d'évaluer les performances de quelques algorithmes d'extraction termes de temps d'exécution et de consommation de l'espace mémoire. Les algorithmes d'extraction et les résultats obtenus font l'objet d'une étude comparative.

Nous pouvons conclure que l'extraction est une tâche difficile et qu'elle nécessite une bonne sélection de moyens et de technologies, y compris le langage de programmation.

Nous regrettons que la recherche soit, complétant le spectre des contraintes de temps.

Et comme perspective, nous proposons d'étendre ce travail à d'autres algorithmes et de faire des études comparatives en fonction d'autres langage.

Et vu que la plupart des données réelles intéressantes dans ce contexte sont numériques (quantitatives), il serait donc intéressant d'introduire la notion.

Reference

1. https://www.coursehero.com/file/84733162/Cours-DataMining-R-seance1pdf/

2. J. Zaki ,SPADE : An efficient algorithm for mining frequent sequences

3. J. Zaki ,SPADE : An efficient algorithm for mining frequent sequences

4.Machine Learning Journal, Vol. 42(1-2), pp. 31-60, January 2001

5. M. Mohammed Nassim./ Extraction des connaissances dans un environnement distribué

6. Marc Plantevit. Extraction De Motifs Séquentiels Dans Des Données Multidimensionelles.

7. Informatique [cs]. Université Montpellier II - Sciences et Techniques du Languedoc, 2008.

8. Français. Fftel00319242f. 7 September 2008 http://www.univ-usto.dz/theses_en_ligne/doc_num.php?explnum_id=669

9.https://fr.wikipedia.org/wiki/R%C3%A8gle_d%27association#:~:text=Dans%20le%20domaine%20du%20data,tr%C3%A8s%20importantes%20bases%20de%20donn%C3%A9es. 18/06/2021

10.Heraguemi, K. E. (2018). *Approche bio-inspirée pour l'extraction des règles d'association* (Doctoral dissertation).

11.https://www.softwaretestinghelp.com/apriori-algorithm/

(Tan et al., KDD 2000; Tan, Steinbach & Kumar, 2006, p. 469)

(Tan et al., 2000)

12.https://www.softwaretestinghelp.com/fp-growth-algorithm-data-mining/

13.https://www.philippe-fournier-viger.com/spmf/Eclat_dEclat.php

14.https://www.philippe-fournier-viger.com/spmf/IndirectAssociationRules.php

15.https://www.philippe-fournier-viger.com/spmf/HMine.php

16.https://fr.wikipedia.org/wiki/Git 19/06/2022

17.https://en.wikipedia.org/wiki/GitHub 19/06/2022

18.https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203597-php-hypertext-preprocessor-definition/ 19/06/2022

19.https://www.philippe-fournier-viger.com/spmf/index.php