ALOA2i : OPTIMISATION D'EXTRACTION DES K-ITEMSETS FREQUENTS (POUR K $\leq$ 2)

Claude Issa Nombré, Konan Marcelin Brou, Kouadio Prosper Kimou

ALOA2i : OPTIMISATION D'EXTRACTION DES K-ITEMSETS FREQUENTS (POUR K ≤ 2)
ALOA2i : OPTIMIZATION OF EXTRACTION K-itemsets FREQUENT (FOR K ≤ 2)

NOMBRÉ Claude Issa, Docteur BROU Konan Marcellin, Docteur KIMOU Kouadio Prosper
Département d'informatique
Ecole Doctorale Polytechnique
Institut National Polytechnique
Houphouet Boigny (EDP-INPHB)
YAMOUSSOUKRO
CÔTE D'IVOIRE
nombreclaude@yahoo.fr

RÉSUMÉ. Dans cet article, nous proposons une nouvelle approche d'optimisation de l'algorithme de référence APRIORI (AGR 94).La démarche utilisée est basée sur des ensembles à un et deux items. Nous commençons par calculer les supports des 1-itemsets (ensembles de singletons), ensuite nous élaguons les 1-itemsets non fréquents et ne conservons que ceux qui sont fréquents (c'est-à-dire ceux qui ont des fréquences d'apparition appelées supports dont les valeurs sont supérieures ou égales à un seuil minimal fixé).Pendant la deuxième itération, nous trions les 1-itemsets fréquents par ordre décroissant de leurs supports respectifs puis nous formons les 2-itemsets. De cette façon les règles d'association sont découvertes plus rapidement.Expérimentalement, la comparaison de notre algorithme avec APRIORI, PASCAL, CLOSE et MAX-MINER, montre son efficacité sur des données faiblement corrélées.
MOTS-CLÉS : Optimisation, Itemsets fréquents, Règles d'association, Données faiblement corrélées, Supports
ABSTRACT. In this article, we propose a novel optimization approach to the reference algorithm APRIORI (AGR 94).The approach used is based on sets one and two items. We start by calculating the supports of 1-itemsets (sets of singletons), then we prune the infrequent 1-itemsets and only keep those that are common (that is to say those with frequencies of occurrence called media whose values are greater than or equal to a minimum threshold). During the second iteration, we sort the frequent 1-itemsets in descending order of their respective holders and then we train 2-itemsets. This way association rules are discovered more quickly. Experimentally, the comparison of our algorithm with APRIORI, PASCAL, CLOSE and MAX-MINER, shows effectiveness to weakly correlated data.
KEYWORDS: Optimization, frequents itemsets, Association Rules, weakly correlated data, Supports.

INTRODUCTION

Au vu du nombre croissant de grandes bases de données et de la complexité des algorithmes mis en œuvre pour son exploitation, la question d'optimisation préoccupe de plus en plus les chercheurs du domaine de la fouille de données.

La découverte des connaissances utiles à partir des données volumineuses est un processus de la fouille de données (Data mining en anglais). L'une des techniques les plus utilisées pour extraire ces connaissances est la méthode des règles d'association. Elle permet de rechercher les similitudes entres les données d'une base de données.

De nombreux travaux et applications ont été proposés. De ces travaux se dégagent deux grands axes de recherche :

- Le premier axe concerne la découverte des règles d'association pertinentes et utiles aux experts d'un domaine donné.
- Le deuxième axe quant à lui s'intéresse à l'optimisation d'extraction de ces règles d'association

Dans ce rapport de thèse, nous nous intéressons en priorité au second axe en développant une autre approche d'optimisation de découverte des 2-itemsets. La génération des règles d'association possédant une prémisse et une conclusion est potentiellement utiles à l'utilisateur final en découle.

Notre travail présente un état de l'art dans le domaine de la découverte des itemsets ou motifs fréquents. Ensuite nous proposons notre travail qui fait l'objet d'un article. Enfin nous concluons par des perspectives.

SECTION 1

1. PRESENTATION DU THEME

1.1. CONTEXTE

Nous avons remarqué que la population ivoirienne, selon la catégorie sociale fait ses achats dans les grandes surfaces en fonction des évènements circonstanciels et des types de produits. On peut classer cette clientèle en deux grandes catégories essentielles :

- Celle qui fait des achats une fois par mois, généralement entre le 28 du mois en cours et le 05 du mois suivant ;
- Celle qui achète par occasion

Fort de ce constat, nous avons décidé de toucher le problème du doigt. Pour cela, nous avions contacté un grand magasin de la place, le CDCI, pour avoir quelques tickets de caisse que nous avons pu rassembler sur une période de trois (03) mois consécutifs, allant du mois de janvier au mois de mars 2016.

En analysant de près ces documents et en interrogeant les responsables du magasin, nous avons pu confirmer la remarque faite plus haut. Car des articles bien précis étaient achetés fréquemment ensemble en fin de mois, de la période du 28 du mois en cours au 05 du mois suivant. A cette période, le chiffre d'affaire du magasin grimpait fortement. Après le 05 et avant le 28 du mois en cours, le magasin enregistrait par jour une baisse relative de sa clientèle, et son chiffre d'affaire baissait de façon significative. Par ailleurs, le responsable du magasin a précisé que des achats plus importants sont fait en période de fêtes. Il faut aussi noter, qu'avec l'autorisation du responsable, nous avons pu visiter la disposition des produits dans le magasin. A l'issu de cette visite, nous avons dressé une cartographie des articles du magasin.En faisant un rapprochement de la disposition des articles avec les tickets de caisse mis à notre disposition, un autre constat surprenant a été fait.

En effet, sur les tickets de caisse on trouvait des articles qui s'achetaient le plus souvent ensemble (constat fait à partir des tickets de caisse), mais qui étaient éloignés physiquement les uns des autres. Nous avons donc conclu que la gestion des magasins ivoiriens en général et en particulier le CDCI de Yamoussoukro que nous avons étudié, n'est pas dynamique. Ainsi, un certain nombre de questions auxquelles nous tenterons de répondre dans la suite notre papier ont été dégagées.

1.2. PROBLEMATIQUE

La problématique de notre thème tourne globalement autour de trois (03) questions essentielles :

- Quels produits peut-on disposer dans le magasin, à quelle période et avec quelle quantité précise ?
- Comment disposer côte à côte les produits qui s'achètent fréquemment ensemble ?

- Quelles stratégies marketing et quelle approche scientifique faut-il mettre en place pour rendre plus efficace la prise de décision optimale, dynamique, exacte et concise des experts du domaine.

1.3. DISCUSSION

Dans cette section, nous avons pris le cas particulier du domaine du marketing du « panier de la ménagère ». Par ailleurs, cala peut s'appliquer également aux autres domaines comme la médecine et le WEB, mais dans des contextes différents.

SECTION 2

2. ETAT DE L'ART SUR LA DECOUVERTE DES ITEMSETS FREQUENTS

2.1. INTRODUCTION

Notre travail est basé sur l'extraction des itemsets fréquents (un itemset est un ensemble d'tems de taille k). Plusieurs travaux ont été réalisés dans ce domaine ouvrant ainsi quatre grands axes de recherches. Les recherches ont depuis deux décennies été orientées sur les deux objectifs principaux :

- L'optimisation d'extraction des itemsets fréquents
- L'extraction des règles d'association (connaissances) pertinentes à partir des itemsets fréquents

Le premier axe de travail utilise la stratégie selon laquelle l'ensemble des itemsets fréquents est parcouru par boucle itérative. A chaque étape k, un ensemble de candidats est créeen joignant les (k-1)-itemsets fréquents. Ensuite, les supports des k-itemsets sont calculés et évalués par rapport à un seuil minimal fixé par l'utilisateur selon son domaine, dans l'objectif de découvrir des k-itemsets fréquents.Les itemsets non fréquents sont supprimés. L'algorithme qui utilise cette stratégie est l'algorithme de référenceApriori [AGR 94], proposé parallèlement à l'algorithme OCD [MAN 94]. Depuis lors des travaux intéressants ont été réalisés pour améliorer un ou plusieurs aspects de cet algorithme [BRI 97b, GAR 98, PAR 95, SAV 95, TOI 96]. Le point commun de tous ces algorithmes est de calculer le support de tous les itemsetssans se préoccuper à priori s'ils sont fréquents ou non.

Une autre approche d'optimisation d'Apriori proposé en 2010 parYves Bastide et al. [BAS 10] dans son article « PASCAL : Une optimisation d'extraction des motifs fréquents. ». Cette méthode est destinée à réduire le nombre de calculs de supports des itemsets lors de l'extraction des itemsets fréquents. Cette méthode repose sur le concept d'itemsets clés. Un motif clé est un motif minimal d'une classe d'équivalence regroupant tous les itemsets contenus dans exactement les mêmes objets de la base de données. Tous les itemsets d'une classe d'équivalence possèdent le même support, et le support des itemsets non clés d'une classe d'équivalence peut donc être déterminé en utilisant le support des itemsets clés de cette classe. Avec le comptage par inférence, seuls les supports des itemsets clés fréquents (et de certains non fréquents) sont calculés depuis la base de données.

Le second axe traite de l'extraction des itemsets fréquents maximaux. Cette stratégie permet de déterminer des ensembles d'itemsetsfréquents (les maximaux) dont aucun de ses supersets immédiats n'est fréquent.

Les algorithmes qui utilisent cette stratégie parcourent les itemsets en combinant la technique d'intelligence artificielle du parcours en largeur d'abord de bas vers le haut et du haut vers le bas. Les itemsets fréquents sont extraits immédiatement après que les itemsets maximaux sont connus.Les algorithmes Max-clique et Max-eclat [ZPOL97], Max-miner [Bay98], Pincer-search [LK98], Depth-Project [AAP00] permettent de parcourir tous les itemsets du treillis (soit 2^nitemsets possibles), mais leurs performances

diminuent quand n augmentedu fait du coût du dernier balayage. L'algorithme le plus efficace basé sur cette approche est l'algorithme Max-Miner.
Le troisième axe traite de la découverte des itemsets fermés fréquents. Cette stratégie utilise la fermeture de la connexion de Galois [GAN 99, DUQ 99]. Un itemset fermé est un ensemble d'items dont le support est différent de tous les supports de ses supersets. Les algorithmes qui utilisent cette stratégie réalisent un balayage par niveaux, tout en découvrant à chaque étape tous les itemsets fréquents et leur support à partir des itemsets fermés fréquents, sans effectuer un accès à la base de données. L'algorithme le plus efficace basé sur cette approche surtout sur des données fortement corrélées est l'algorithme Close [PAS 99c],
Le quatrième axe est basé sur une méthode hybride utilisant les deux premières approches. Les algorithmes adoptant cette stratégie explorent l'espace de recherche en profondeur d'abord tel que les algorithmes de la deuxième méthode souvent appelée « Diviser-et-générer »,mais sans diviser le contexte d'extraction en des sous-contextes. Ces algorithmes génèrent un ensemble de candidats tel que c'est le cas dans la première méthode appelée souvent « Générer-et-tester ». Cependant, cet ensemble est toujours réduit à un seul élément. Ces derniers consistent à utiliser une métrique statistique en conjonction avec d'autres heuristiques.
En outre Il existe d'autres travaux basés sur la stratégie hybride. Ces travaux ont permis de combiner des techniques de datamining entre elles ou entre des techniques de datamining et des méthodes probabilistes comme les réseaux bayésiens. Quelques travaux intéressants ont effectués dans cet axe de recherche :

- Les travaux de Nicolas Voisine et al. (2009) ont présenté un nouvel algorithme automatique pour l'apprentissage d'arbres de décision. Ils abordent le problème selon une approche Bayésienne en proposant, sans aucun paramètre, une expression analytique de la probabilité d'un arbre connaissant les données. Enfin ils transforment le problème de construction de l'arbre en un problème d'optimisation.
- Les travaux d'Eliane Tchiengue (2011) ont permis de proposer dans un projet NICE Datamining du système d'information du Crédit Agricole français, avec pour objectif d'apporter des solutions pour la mise en place d'un outil d'analyse de données et de modélisation datamining qui optimise la connaissance client.
- L'Algorithme Microsoft Association (2016) est un algorithme qui est souvent utilisé pour les moteurs de recommandation. Un moteur de recommandation recommande des articles à des clients basés sur les articles qu'ils ont déjà achetés, ou dans lequel ils ont manifesté leur intérêt. L'algorithme Microsoft Association est également utile pour l'analyse du panier du marché.
- Jaroszewicz et Simovici (2004); Jaroszewicz et Scheffer (2005) décrivent l'utilisation d'un réseau bayésien pour calculer l'intérêt d'ensembles d'attributs extraits à l'aide d'un algorithme de type Apriori.
- Clément Fauré (2007, 2014) montre l'intérêt de la mise en œuvre de réseaux Bayésiens couplée à l'extraction de règles d'association dites delta-fortes, où

l'utilisateur (expert) est au centre de la découverte des règles d'association potentiellement utiles est facilité par l'exploitation des connaissances décrites par l'expert et représentées dans le réseau Bayésien.

2.2. CONTRIBUTION

Notre hypothèse de travail prend en compte les ensembles d'itemsets à un et deux items. Les contributions de notre article se déclinent en deux points. Tout d'abord, nous tentons de résoudre un problème qui se pose aux décideurs qui ont souvent du mal à choisir les articles à disposer ensemble dans une grande surface par exemple. Le deuxième point consiste à améliorer de façon significative la génération des 2-itemsets fréquents.Le troisième point découle du deuxième par la découverte de façon concise et pertinente des règles d'association à deux items.

2.3. CONCLUSION PARTIELLE

Dans notre état de l'art, nous avons présenté quatre catégories d'algorithmes des règles d'association basées sur des itemsets fréquents, maximaux, fermés et hybrides sur des données fortement et faiblement corrélées. Les sur ensembles d'itemsets fréquents ayant des supports faibles par rapport aux sous-ensembles d'itemsets fréquents, la probabilité de trouver des règles d'association intéressantes est plus élevée avec les ensembles à items réduits. C'est une raison de plus qui nous a guidé dans notre choix de travailler avec des itemsets de tailles k=1 et k=2. Les 1-itemsets ne peuvent pas être des règles, mais les 2-itemsets peuvent produire des règles d'association beaucoup intéressantes et très pertinentes.

SECTION 3

3. ORGANISATION DU TRAVAIL

3.1. APPROCHE METHODOLOGIQUE

Notre approche s'applique uniquement sur des ensembles à un et deux items avec deux niveaux (i=1 et i=2), Ce choix se justifie aisément par les éléments suivants :

- Etant donné qu'une règle d'associationest constituée d'au moins de deux items de la forme a→b (a est la prémisse et b la conclusion), nous ne pouvons pas considérer uniquement les itemsets à un seul cardinal ;
- Le tempsd'extraction des itemsets fréquents de cardinaux k (avec $\mathrm{k} = \overline{1,\mathrm{n}}$), dépend de la somme des temps écoulés au cours de l'exécution de chacune des itérations (complexité). Le temps mis pour exécuter chaque itération dépend du nombre d'itemsets de même cardinal. Supposons une base de données D, contenant k=6 items. Vérifions le nombre d'itemsets par cardinal identique :

Cardinal n = 1, k = 6	$C_6^1 = \frac{6!}{1!5!} = 6$
Cardinal n = 2, k = 6	$C_6^2 = \frac{6!}{2!4!} = 15$
Cardinal n = 3, k = 6	$C_6^3 = \frac{6!}{3!3!} = 20$
Cardinal n = 4, k = 6	$C_6^4 = \frac{6!}{4!2!} = 15$
Cardinal n = 5, k = 6	$C_6^5 = \frac{6!}{5!1!} = 6$
Cardinal n = 6, k = 6	$C_6^6 = \frac{6!}{6!0!} = 1$

Tableau 1 : Le nombre d'itemsets par cardinal n=1 à 6

Nous constatons, même pour un k supérieur à 6 que le nombre d'ensembles d'itemsets de cardinal 2 est parmi les plus élevés, surtout lorsque k n'est pas très grand. La fonction factorielle étant symétrique par rapport à $C_k^{k/2}$ ($\mathrm{n} = \mathrm{k}/2$), nous pouvons donc dire qu'optimiser les itemsets fréquents de cardinal 2 constitue une avancée majeure dans l'optimisation globale de tous les itemsets de cardinaux k (avec $\mathrm{k} = 1,\mathrm{n}$),

- En outre, l'un des objectifs spécifiques de notre papier serait d'aider l'expert du domaine à prendre de bonnes décisions. Par exemple, celle qui consiste à disposer côte à côte les articles qui apparaissent fréquemment ensemble sur les tickets de caisse. Cette décision peut justifier aussi notre choix de limiter les ensembles d'itemsets à 2 items au maximum.

La problématique de notre thème répond à la question suivante : Comment améliorer les temps d'extraction des 2-itemsets fréquents tout en découvrant des connaissances utiles à une prise de décision ? Plusieurs travaux ont déjà été réalisés dans le domaine de la découverte des itemsets fréquents et des règles d'association utiles, comme nous l'avions indiqué tantôt dans l'état de l'art.

Cette partie s'intéresse à la démarche à suivre pour trouver une solution à cette question. Tout comme ARIORI, notre approche commence par calculer avec un accès à la base de données, les supports des 1-itemsets (ensemble à un item). Sans un autre accès à la base de données on détermine les supports des 2-itemsets.

Notre démarche se décrit en ces étapes :

- Explorer les 1-itemsets puis calculer leurs supports afin d'élaguer les non fréquents (supports inférieurs à un seuil minimal fixé)
- Classer les 1-itemsets fréquents par ordre décroissant des supports
- Générer les 2-itemsets fréquents, puis calculer leurs supports.
- Découvrir les règles d'association exactes.

3.2. LES OUTILS DE TRAVAIL

Nos outils de travail passent d'abord par une revue de littératurequi nous a permis de ressortir un état de l'art sur la découverte des itemsets fréquents. Nous avons utilisé également des propositions basées sur des notions mathématiques afin de consolider notre travail, des techniques d'intelligence artificielle et de datamining.

La technique de data mining utilisée ici est la méthode répandue des règles d'association. La phase de réalisation de notre travail repose sur l'algorithme d'optimisation d'extraction des 2-itemsets de l'algorithme de référence APRIRI que nous appelons ALOA2i (ALgorithme d'Optimisation de l'algorithme Apriori avec 2-itemsets).

3.3. PHASE THEORIQUE

Définition1. Base de données

Soit O un ensemble fini d'objets, P un ensemble fini d'éléments ou items, et R une relation binaire entre ces deux ensembles. On appelle base de données ou contexte formel [GAN 99] le triplet D = (O,P,R). La base de donnés D représente notre espace de travail

Définition 2 : Transaction

Soit $T = \{T_1, T_2, T_3, \ldots, T_n\}$, $T_i \subseteq D$. On appelle T_i un ensemble de lignes contenant les occurrences de la base de données D. Tous les Tisont appelés des transactions. Dans l'exemple connu du panier de la ménagère, les transactions sont les tickets de caisse (c'est-à-dire les achats effectués par les clients).

Définition 3 : Item

On appelle item toute variable X_i représentant une occurrence de D

Définition 4 : Itemset

On appelle itemset, l'ensemble formé d'items. Exemple le singleton $\{X_1\}$ et la paire $\{X_1,X_2\}$ sont des itemsets. Un itemset de taillek est noté k-itemset

Définition 5 : Support
On appelle support le pourcentage de Ti où apparait la règle d'association, c'est à dire

$$Support(Xi \rightarrow Xj) = \frac{Freq(Xi \cup Xj)}{Card(T)}$$

$$ou\ Support(Xi \rightarrow Xj) = Nbre\ de\ fois\ où\ Xi\ et\ Xj\ apparaissent\ ensemble\ dans\ les\ transactions\ T$$

Définition 6 : Confiance

On appelle confiance le pourcentage de fois où la règle est vérifiée, c'est-à-dire

$$Confiance(Xi \rightarrow Xj) = \frac{Freq(Xi \cup Xj)}{Freq(Xi)}$$

Définition 7 : Superset
Un superset est un itemset défini par rapport à un autre itemset.
Exemple {a,b,c} est un superset de {a, b}.
Définition 8 : Itemset fréquent
Un Itemset fréquent est un itemset dont le support est ≥ à minsup (support minimal en dessous duquel l'itemset est considéré non fréquent). Si un itemset n'est pas fréquent, tous ses supersets ne le seront pas non plus. Si un superset est fréquent alors tous ses sous itemsets sont aussi fréquents (propriété anti-monotone)
Définition 9 : Itemset fermé
Un itemset fréquent est dit fermé si aucun de ses supersets n'a de support identique. Autrement dit, tous ses supersets ont un support strictement plus faible.
Définition 10 : Itemset libre
Un itemset est libre s'il n'est pas inclus dans la fermeture d'un de ses ensembles stricts
Définition 11 : Itemset maximal
Un itemset est dit maximal si aucun de ses supersets n'est fréquent.
Définition 12 : Itemset générateur
Un itemset est dit générateur si tous ses sousitemsetsont un support strictement supérieur.
Définition 13 : Support partiel
On dit qu'un support est partiel lorsqu'à l'intérieur d'une transaction, deux items possèdent la valeur égale à 1. C'est à dire que la transaction contient deux attributs de la base binaire dont les valeurs sont égales à 1 Exemple : si a=1 et b=1, alors le support partiel de la règle a→b est égal à 1.

Proposition 1
Soit D, une base de données et Ti les transactions de D. Si un attribut Xi n'apparaît qu'une fois ou nullement dans les transactions Ti, alors les règles qui le contiennent en conclusion auront des supports et des confiances très faibles tendant vers 0.

Proposition 2
Une règle est considérée inintéressante lorsque l'une au moins des trois conditions suivantes existent :

1. Xk (k = 1, 2, …, i, i+1,…, j) $\subseteq$ D apparaît une seule fois ou n'apparaît jamais dans Ti $\Rightarrow \cap T_i = \varnothing$ ou égal au singleton.
2. Support (Xi $\rightarrow$Xj) <minSup et Conf (Xi $\rightarrow$Xj) <minConf
3. Support (Xj$\rightarrow$ Xi) <minSup et Conf (Xj$\rightarrow$ Xi) <minConf

Proposition 3 (Transitivité)
Soient deux ensemble I = {I_1, I_2, …, I_t}, T = {T_1, T_2, …,T_n} inclus dans la base de données K tel que I $\subseteq$ T. Si les règles d'association : $I_t \rightarrow I_{t+1}$ et $I_{t+1} \rightarrow I_{t+2}$ ont des supports partiels égal à 1, alors par la relation transitive la règle d'association $I_t \rightarrow I_{t+2}$ aura un support partiel égal à 1.

3.4. PRESENTATION DE L'ALGORITHME ALOA2I

Les notations utilisées sont présentées dans le tableau 2 et le pseudo code dans les algorithmes nommés respectivement ALOA2i (**AL**gorithme d'**O**ptimisation d'**A**priori pour les **2-i**temsets) et ALOA2i-gen.

Données	Types	E/S	Assignation
K	Entier	Entrée	Numéro de l'itération en cours
Ck	Matrice d'entiers	Sortie	Candidats de taille k
Fk	Matrice d'entiers	Sortie	Itemsets fréquents de taille k
Minsup	Réel	Entrée	Seuil minimal de support
Minconf	Réel	Entrée	Seuil minimal de confiance
T	Vecteur d'entiers	Sortie	Ensemble de transactions
Support	Réel	Sortie	Support = Freq(k-itemsets)/nbre de transactions
Confiance	Réel	Sortie	Confiance = Support(k-itemsets)/support(prémisse)

Tableau 2. ALOA2i : Annotations

Algorithme 1 : ALOA2i

Entrée : D – Base de données
Minsyp, Minconf réel
K entier
Sortie : Ck – itemsets candidats, Fk – Itemsets fréquents
Support, Confiance réel
T – transactions (t $\subset$ T)

Début

K ← 1
Ck←candiadats 1-itemsets
Fk←ϕ
Pour Chaque transaction t ∈ D Faire
SiSupport.t.items≥Minsup
AlorsFk←Ck∪Fk
Sinon Supprimer Ck.t.itemsets
FSi
ALOA2i_Gen(Fk)
Fpour
Trier par ordre décroissant t.itemsets fréquents ⊆ D
{Les 1-itemsets fréquents triés}
{Formons les 2-itemsets fréquents}
K ← 2
Si (t.itemsets = 1) et ((t+1).itemsets = 1)
Alorsrègle (t.itemsets) → ((t+1).itemsets)← 1
Sinon **Si** (t.itemsets = 1) et ((t+1).itemsets = 0)
ou (t.itemsets = 0) et ((t+1).itemsets = 1)
Alors règle (t.itemsets) → ((t+1).itemsets) ← 0
{Transitivité}
Sinon **Si** (t.itemsets = 1) et ((t+1).itemsets = 1)
et (t+2).itemsets = 1)
Alors règle (t.itemsets) → ((t+2).itemsets) ← 1
FSi
FSi
FSi
SiSupport(t.itemsets) ≥Minsup
AlorsFk←Ck∪Fk
Sinon**Supprimer**Ck.t.itemsets
FSi
ALOA2i_Gen(Fk)
FPour
Retourner∪kFk
Fin

Algorithme 2 : ALOA2i_Gen

Entrée : t.itemsets
Sortie : t.candidat, Ck

Début

Pour Chaque pairs d'itemsets
t.candidat←t.itemsets∪ (t+1).itemsets

Ck←Ck∪t.candidat
FPour
RetournerCk
Fin

Pour comparer théoriquement notre méthode avec Apriori sur k niveaux, avec k=1,2, nous considérons l'exemple suivant dont les données sont issues de 6 tickets de caisse que nous avons obtenu avec 6 clients du supermarché CDCI YAMOUSSOUKRO (Côte d'Ivoire) :

Sot BD la base de donné7es binaire des transactions T et K = {A,B,C,D,E,F,G,H,I,J}

t_1→ {Lait, biscuit, savon, riz, sucre}
t_2→ {Cannette de bière, huile, rasoir blue II}
t_3→ {Savon, huile, cannette de bière, sucre}
t_4→ {Spaghetti, huile, rasoir blue II}
t_5→ {Lait, biscuit, sucre, boîte de nescafé}
t_6→ {savon, riz, sucre, huile}

A →Lait B → Biscuit C → Savon D → Riz E → Sucre F → Cannette de bière G → Huile H → Rasoir blue 2 I → Spaghetti J → Nescafé

Le tableau suivant représente la base de données des transactions, où chaque transaction est une liste des articles achetés par l'un des 6 clients du supermarché :

Ti	A	B	C	D	E	F	G	H	I	J
t_1	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
t_2	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**0**
t_3	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**0**	**0**
t_4	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**0**
t_5	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**
t_6	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**0**

Figure 1 : liste des produits

- Testons cet exemple avec l'algorithme APRIORI avec minsup = 2/6

Base de Données K

Ti	Items
1	{A,B,C,D,E}
2	{C,E,F,G}
3	{C,E,F,G}
4	{G,H,I}
5	{A,B,E,J}
6	{C,D,E,G}

Candidats C1

Ti	Itemsets	Support
1	{A}	2/6
2	{B}	2/6
3	{C}	3/6
4	{D}	2/6
5	{E}	4/6
6	{F}	2/6
7	{G}	4/6
8	{H}	2/6
9	{I}	1/6
10	{J}	1/6

1-Itemsets fréquents F1

Ti	1-Itemsets	Support
1	{A}	2/6
2	{B}	2/6
3	{C}	3/6
4	{D}	2/6
5	{E}	4/6
6	{F}	2/6
7	{G}	4/6
8	{H}	2/6

1-itemsets élagués : {I} et {J}

Candidats C2

Ti	2-Itemsets	Support
1	{A,B}	2/6
2	{A,C}	1/6
3	{A,D}	1/6
4	{A,E}	2/6
5	{A,F}	0/6
6	{A,G}	0/6
7	{A,H}	0/6
8	{B,C}	1/6
9	{B,D}	1/6
10	{B,E}	2/6
11	{B,F}	0/6
12	{B,G}	0/6
13	{B,H}	0/6
14	{C,D}	2/6
15	{C,E}	2/6
16	{C,F}	1/6
17	{C,G}	3/6
18	{C,H}	0/6
19	{D,E}	2/6
20	{D,F}	0/6
21	{D,G}	1/6
22	{D,H}	0/6
23	{E,F}	1/6
24	{E,G}	2/6
25	{E,H}	0/6
26	{F,G}	2/6
27	{F,H}	1/6
28	{G,H}	2/6

2-Itemsets fréquents F2

Ti	2-Itemsets	Support
1	{A,B}	2/6
2	{A,E}	2/6
3	{B,E}	2/6
4	{C,D}	2/6
5	{C,E}	2/6
6	{C,G}	3/6
7	{D,E}	2/6
8	{E,G}	2/6
9	{F,G}	2/6
10	{G,H}	2/6

APRIORI a généré en tout 18 itemsets fréquents (huit 1-itemsets fréquents et dix 2-itemsets fréquents). Calculons les confiances des règles d'association à 2 items.

Si nous fixons minconf = 50% nous constatons que toutes les règles sont valides. Cependant observons-les de plus près. Les règles A→ B, B → E et A → E sont redondantes car elles se résument toutes à la règle A → E. Il en est de même pour les règles C→D, C→ E et D→ E qui se résument à C →E. De cette façon, nous en déduisons que 4 règles sont redondantes. D'où en retranchant les quatre 2-itemsets redondants, nous obtenons véritablement six 2-itemsets utiles. Nous montrons que l'algorithme Apriori perd du temps inutile à calculer les supports des règles d'association qui ne serviront pas en réalité à l'utilisateur final.

- Testons maintenant l'exemple avec notre algorithme avec minsup = 2/6

Base de Données K

Ti	Items
1	{A,B,C,D,E}
2	{C,E,F,G}
3	{C,E,F,G}
4	{G,H,I}
5	{A,B,E,J}
6	{C,D,E,G}

Candidats C1

Ti	Itemsets	Support
1	{A}	2/6
2	{B}	2/6
3	{C}	3/6
4	{D}	2/6
5	{E}	4/6
6	{F}	2/6
7	{G}	4/6
8	{H}	2/6
9	{I}	1/6
10	{J}	1/6

1-Itemsets fréquents F1

Ti	1-Itemsets	Support
1	{E}	4/6
2	{G}	4/6
3	{C}	3/6
4	{A}	2/6
5	{B}	2/6
6	{D}	2/6
7	{F}	2/6
8	{H}	2/6

1-itemsets élagués : {I} et {J}

Candidats C2

Ti	2-Itemsets	Support
1	{E,G}	2/6
2	{G,C}	2/6
3	{C,A}	1/6
4	{A,B}	2/6
6	{B,D}	1/6
7	{D,F}	1/6
8	{F,H}	1/6

2-Itemsets fréquents F2

Ti	2-Itemsets	Support
1	{E,G}	2/6
2	{G,C}	2/6
3	{A,B}	2/6

Mettons en prémisse (partie gauche) dans une règle l'item de support maximal (support le plus élevé entre les deux items d'une règle). Puis utilisons la propriété de transitivité

(propriété 1) et à l'aide de F1 afin de découvrir les règles d'association qui découlent des 3 règles de F2. Ainsi F2 devient:

Ti	2-Itemsets	Support
1	{E,G}	2/6
2	{G,C}	2/6
3	{E,C}	2/6
4	{E,A}	2/6
5	{A,B}	2/6
6	{E,D}	2/6

Comparativement à APRIORI notre algorithme produit moins de règles d'association à deux items, car les règles d'association redondantes produites dans l'algorithme APRIORI sont automatiquement supprimées.

Notons que pendant la deuxième étape (k=2) l'espace de recherche est considérablement réduit grâce aux sauts opérés lorsqu'il existe un zéro (0) entre deux valeurs d'items.
Fixons minconf = 50%

Ti	2-Itemsets	Support	Confiance
1	{E,G}	2/6	½ = 50%
2	{G,C}	2/6	1 = 100%
3	{E,C}	2/6	½ = 50%
4	{E,A}	2/6	½ = 50%
5	{A,B}	2/6	1 = 100%
6	{E,D}	2/6	½ =50%

Nous pouvons conclure que le nombre de règles d'association valides diminue considérablement avec notre méthode par rapport à Apriori.
Pour que ces règles d'association soient valides, il faut qu'elles respectent les trois conditions suivantes :

1. Leurs supports ≥minsup ;
2. Leurs confiances ≥minconf ;
3. Les règles sont auto-réciproques, c'est-à-dire confiance(X→Y) = confiance(Y→X)

Proposition 4
Soit K une base de données et deux items X et Y de K. Si confiance(X→Y) égale confiance(Y→X) alors la règle association X→Y est dit auto-réciproque et support(X) égal support(Y).
Exemple : E→G est une règle d'association auto-réciproque car Confiance(E→G)=Confiance(G→E), mais elle doit en plus satisfaire aux conditions 1 et 2 pour être une règle valide.

SECTION 4

4. EXPERIMENTATIONS

Nos expériences sont essentiellement portées sur des données corrélées et des données faiblement corrélées. Les itemsets fréquents générés sont de tailles k=1 et k=2. Nous nous limiteront à comparer ALOA2i avec Apriori et Pascal, car les travaux de Yves Bastide publié dans l'article « PASCAL : un algorithme d'extraction des motifs fréquents », des expériences ont montré que l'algorithme Pascal a optimisé Apriori avec des temps de réponse bien souvent meilleurs aux algorithmes Close (pour les motifs fréquents fermés), Max-miner (pour les motifs fréquents maximaux) et Apriori (pour les motifs fréquents). Itemsets fréquents et motifs fréquents veulent dire la même chose.

Données faiblement corrélées

– **Jeu de données T20I6D100K**

Support	Fréquents	ALOA2i	Pascal	Apriori
1	192	1,06	1,64	1,69
0,75	589	1,97	2,55	2,58
0,5	3 369	4,92	5,50	5,55
0,25	19 459	14,17	14,75	14,72

Tableau 3 : Temps de réponse pour T20I6D100K

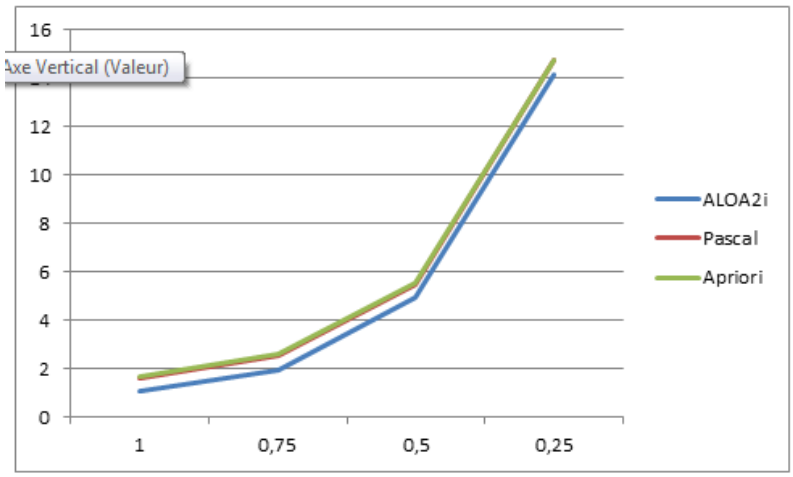

Figure 2 : Les résultats expérimentaux pour T20I6D100K

– **Jeu de données T25I20D100K**

Support	Fréquents	ALOA2i	Pascal	Apriori
1	73	0,06	0,64	0,72
0,75	144	0,64	1,22	1,39
0,5	159 907	120,50	121,08	116,89

Tableau 4 : Temps de réponse pour T25I20D100K

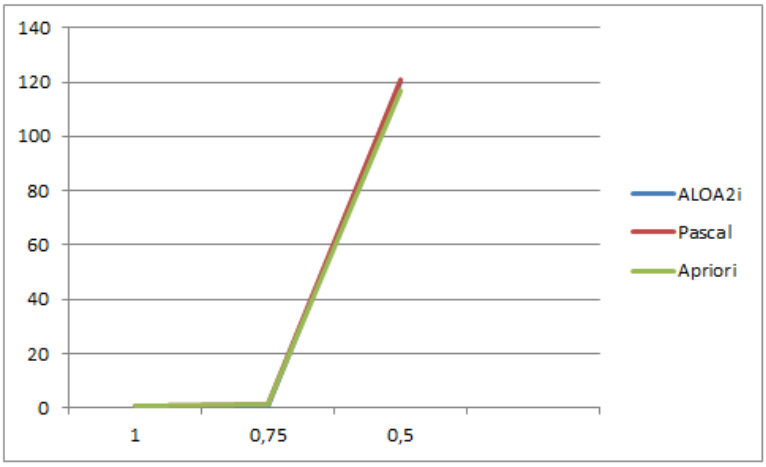

Figure 3 : Les résultats expérimentaux pour T25I20D100K

On constate clairement que les temps de réponse de notre algorithme ALOA2i sont meilleurs aux temps de réponse des algorithmes Pascal et Apriori

Données corrélées

– **Jeu de données C20D10K**

Support	Fréquents	ALOA2i	Pascal	Apriori
20	2 530	3,38	1,18	7,14
15	4 545	5,17	1,54	10,67
10	11 235	12,83	2,41	20,60
7,5	19 145	14,36	2,94	29,05
5	44 076	22,55	4,13	49,42
2,5	145 045	35,34	6,92	94,33

Tableau 5 : Temps de réponse pour C20D10K

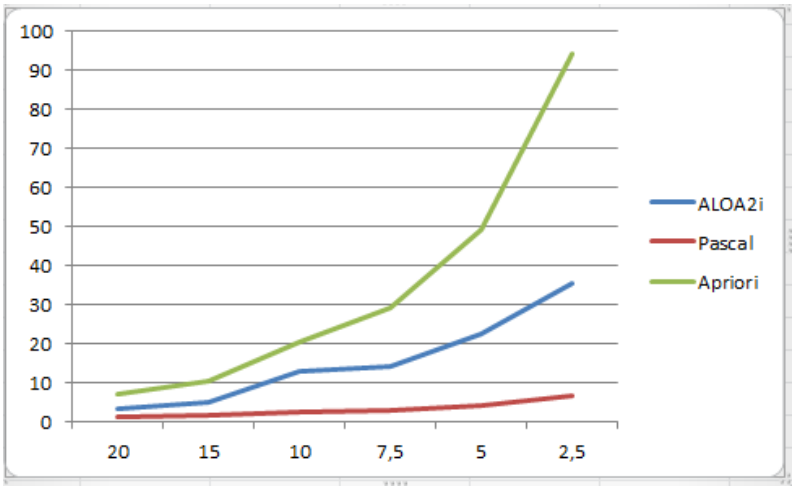

Figure 4 : Les résultats expérimentaux pour C20D10K

– **Jeu de données C73D10K**

Support	Fréquents	ALOA2i	Pascal	Apriori
80	13 645	54,61	22,19	457,66
75	29 409	126,52	49,10	956,70
70	71 511	193,73	98,31	2 183,14
60	544 443	724,93	496,51	13 650,50

Tableau 6 : Temps de réponse pour C73D10K

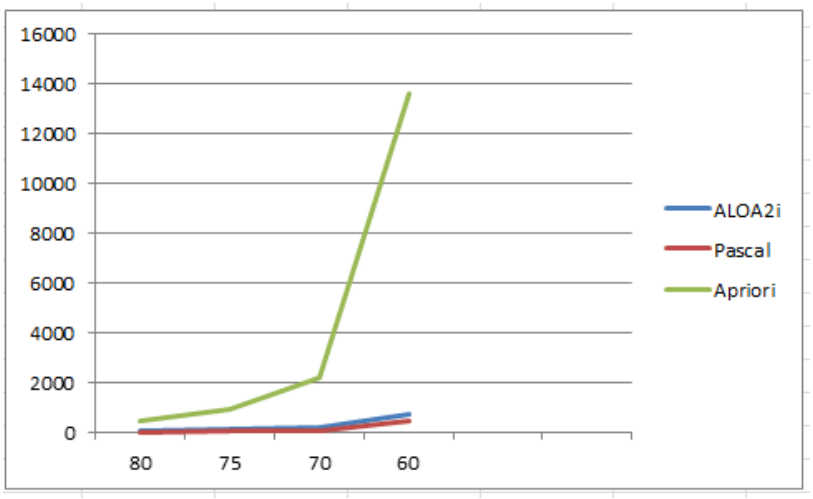

Figure 5 : Les résultats expérimentaux pour C73D10K

Sur les données corrélées ALOA2i donne des temps de réponse supérieurs à l'algorithme Pascal, mais inférieurs à l'algorithme Apriori. Soit T_{aloa2i} – Temps de réponse de l'algorithme ALOA2i; $T_{apriori}$ – Temps de réponse de l'algorithme Apriori et T_{pascal} - Temps de réponse de l'algorithme Pascal. Tpascal≤ Taloa2i ≤Tapriori

SECTION 5

5. CONCLUSION ET PERSPECTIVES

Dans ce papier, nous avons proposé une nouvelle approche d'optimisation d'extraction des 2-itemsets fréquents. Elle nous a permis d'améliorer les temps obtenus dans les travaux antérieurs de découverte des ensembles pairs (2 items) sur des données faiblement corrélées. Notre méthode est certes intéressante, car elle est originale, mais elle est limitée dans un contexte très spécifique. Une première perspective des prochains travaux concerne l'optimisation d'extraction des itemets de cardinal supérieur à 2 à partir de nos résultats et d'améliorer le temps global d'extraction des k-itemsets fréquents. Une autre perspective pourrait envisager d'étendre nos expérimentations à des données denses et éparses. Une dernière perspective pourrait concerner l'amélioration de notre algorithme afin d'obtenir un temps plus court d'extraction des 2-itemsets.

Bibliographie

BIBLIOGRAPHIE

1. R. Agrawal, R. Srikant, H, "Fast algorithms for mining association rules in large databases",Research Report RJ 9839, IBM Almaden Research Center, San Jose, California,June 1994.
2. J. Azé. Extraction de connaissances à partir de données numériques et textuelles. Thèse de doctorat, Université Paris-Sud, december 2003.
3. Boulicaut J-F., A. Bykowski, and C. Rigotti. Approximation of frequency queries by means of free-sets. In : Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases PKDD'00, Lyon (F), September 13-16, 2000.Springer-Verlag LNAI 1910, pp. 75-85.
4. Martine CADOT. Extraire et valider les relations complexes en sciences humaines : statistiques, motifs et règles d'association, Thèse de doctorat de l'université de Franche-Comté – Besançon12 décembre 2006.
5. Yves Bastide et al. Pascal : un algorithme d'extraction des motifs fréquents, Article de doctorat, IRISA-INRIA - 35042 Rennes Cedex, 26 Avril 2010.
6. Ming-Chang Lee. Data mining – R - association rules and apriori algorithm 29 Mars 2009
7. Clément Fauré. Découvertes de motifs pertinents par l'implémentati6n Bayésien : Application à l'industrie aéronautique. Thèse de doctorat. Année 2007
8. Sadok Ben Yahia-EngelbertMephuNguifo. Approches d'extraction de règlesd'association basées sur la correspondance de Galois. Centre de Recherche en Informatique de Lens - IUT de LensRue de l'Université SP 16, F-62307 Lens cedex
9. Thierry Lecroq. Extraction de règles d'association, Université de Rouen France
10. N. Pasquier Y. Bastide R. Taouil et L. Lakhal. Pruning closed itemset lattices for association rules. In Actes des 14^{e} journées Bases de Données Avancées (BDA'98), pages 177{196, 1998.
11. Mostafa El Habib Daho et Al..Dynamic Pruning for Tree-based Ensembles. pages 261. Année 2016.
12. Dr Brou Konan Marcellin. Support de cours intitulé« Chapitre 2 : Règles d'association » INPHB, 2015 – 2016.