

UNIVERSITÉ D'ANGERS

M2 INTELLIGENCE DÉCISIONNELLE

Recherche de motifs fréquents par algorithmes évolutionnaires

Auteur :
Ugo RAYER

Encadrants :
Benoit DA MOTA
Béatrice DUVAL
David LESAINT

10 mars 2017



Remerciements

Avant toute chose, je tiens à remercier l'ensemble des personnes qui ont participé, de près ou de loin, à la réalisation de ce stage et à l'écriture de ce rapport.

Des remerciements spéciaux sont adressés d'une part au Laboratoire d'Etude et de Recherche en Informatique d'Angers pour son accueil et d'autre part au projet GRIOTE de la région Pays de la Loire qui a financé cette étude. Ensuite, je tiens à remercier chaleureusement Madame Duval et Messieurs Da Mota et Lesaint pour la qualité de leur encadrement et les différentes remarques qu'ils m'ont prodiguées pendant ces quelques mois.

Enfin, un grand merci à Josépha pour ses précieux conseils d'écriture malgré l'incompréhension générale des sujets abordés.

Table des matières

1	Introduction	4
2	Le problème du calcul motifs fréquents	6
2.1	Définition du problème	6
2.1.1	Formalisation	6
2.1.2	Exemple	7
2.1.3	Complexité et Propriétés	7
2.2	Problème des itemsets clos maximaux	8
3	Etat de l'art	10
3.1	Méthodes exactes	10
3.1.1	A Priori	10
	Bibliographie	11

Chapitre 1

Introduction

Le calcul des motifs fréquents est une notion essentielle dans de nombreux domaines liés à la découverte et l'extraction de connaissances. Initialement introduit par Agrawal et al. dans [1], ces motifs étaient alors utilisés pour l'établissement de règles d'associations visant à caractériser les habitudes d'achats de clients d'un supermarché. Par exemple, une règle de la forme "Pain & Beurre \Rightarrow Jambon (75%)" signifie que 3 clients sur 4 achetant du pain et du beurre achètent également du jambon. Le calcul de telles règles se fait en deux étapes, dont la principale (i.e. disposant de la plus grosse complexité calculatoire) correspond au calcul des motifs fréquents. Depuis son introduction, le problème du calcul des motifs fréquents a été très largement étudié et appliqué à de nombreux domaines comme en bio-informatique, en cybersécurité et bien évidemment en marketing.

L'avènement de l'ère du Big Data a fait rentrer le problème de calcul des motifs fréquents dans une nouvelle dimension. En effet, le volume de données produites dans tous les domaines a cru de manière vertigineuse ces dernières années, rendant l'extraction de connaissances d'autant plus nécessaire et délicate. De fait, la problématique du passage à l'échelle des méthodes exactes est devenue primordiale. Bien que divers efforts en ce sens aient été faits au travers de nombreuses publications, ils se concentrent généralement sur l'optimisation et la parallélisation des méthodes existantes.

Les algorithmes évolutionnaires font partie des techniques de résolution de problèmes combinatoires appelées méta-heuristiques. Les méta-heuristiques regroupent un ensemble de méthodes approchées à la résolution de problème combinatoire. Elles sont naturellement envisagées lorsque la complexité du problème étudié ne permet pas l'usage de méthodes exactes (aussi bien en temps qu'en espace). Le principe des algorithmes évolutionnaires est de manipuler un ensemble d'individus représentant chacun une solution au problème étudié. A chaque itération, certains individus sont modifiés via des opérateurs de croisement et de mutation. Chaque individu est évalué au regard d'une fonction à optimiser dépendant du problème étudié.

Ainsi, nous formaliserons le problème de calcul de motifs fréquents dans la section suivante avant d'effectuer un état de l'art des méthodes existantes en

section 3. Le chapitre 4 sera dédié à la présentation de notre méthode dont nous présenterons les résultats en section 5. Le dernier chapitre sera consacré à la conclusion de cette étude et à une ouverture vers de futurs travaux.

Chapitre 2

Le problème du calcul motifs fréquents

2.1 Définition du problème

La définition la plus courante du problème de calcul des motifs fréquents se fait par la théorie des ensembles. Nous verrons cependant dans le chapitre suivant qu'il peut également être décrit par la théorie des graphes. Nous présenterons dans un premier temps un cadre formel nécessaire à la définition du problème que nous illustrerons ensuite. Enfin, nous introduirons quelques propriétés dérivées de la définition du problème.

2.1.1 Formalisation

Soit \mathcal{I} un ensemble de *symboles* appelées **items**. Quelque soit $I \subseteq \mathcal{I}$, I est un *motif* appelé **itemset**.

Soit $\mathcal{T} = \{ t_1, \dots, t_n \}$ un ensemble de **transactions**. Chaque élément t_i est un couple $\langle tid, I \rangle$ où tid est l'identifiant de la transaction et $I \subseteq \mathcal{I}$. \mathcal{T} est communément appelé **base de transactions**.

Pour tout itemset $I \subseteq \mathcal{I}$ la **couverture** de I par \mathcal{T} est définie par :

$$\mathbf{cover}_{\mathcal{T}}(I) = \{ t \in \mathcal{T} \mid I \subseteq t \}$$

La cardinalité de la couverture d'un itemset I par \mathcal{T}

$$\mathbf{sup}_{\mathcal{T}}(I) = | \mathbf{cover}_{\mathcal{T}}(I) |$$

est appelée **support** de I . Etant donné un support minimal *minsup* l'ensemble des **itemsets** (i.e. motifs) **fréquents** est défini par :

$$\mathbf{F}_{\mathcal{T}, \text{minsup}} = \{ I \subseteq \mathcal{I} \mid \mathbf{sup}_{\mathcal{T}}(I) \geq \text{minsup} \}$$

Le problème du **calcul des itemsets fréquents** (**FIM** - *Frequent Itemsets Mining*) est, étant donné une base de transactions \mathcal{T} et un support minimal $minsup$ de calculer l'ensemble F des itemsets fréquents. Comme F. Boden le remarque dans [2], bien qu'historiquement définie comme une valeur relative et donc asujettie à un support minimal défini dans l'intervalle $[0,1]$, le support est de nos jours mesuré de manière absolue. Si nécessaire, nous y ferons référence sous la notion de fréquence :

$$\mathbf{Freq}_{\mathcal{T}}(I) = \frac{|\mathbf{cover}_{\mathcal{T}}(I)|}{|\mathcal{T}|}$$

avec $minfreq$ simplement définie par $\frac{minsup}{|\mathcal{T}|}$.

2.1.2 Exemple

Situons nous dans le contexte de la définition historique de problème et considérons la base de transactions suivante (que nous conserverons tout au long de cet article). Le tableau 1 décrit chaque transaction par : un identifiant, une liste d'objets achetés et la liste des items fréquents vis à vis d'un support minimal de 3.

ID transaction	Objets achetés	Items Fréquents Ordonnés
100	f, c, a, d, g, i m, p	f, c, a, m, p
200	a, b, c, f, l, m, o	f, c, a, b, m
300	b, f, h, j, o	f, b
400	b, c, k, s, p	c, b, p
500	a, f, c, e, l, p, m, n	f, c, a, m, p

Une fois calculé, l'ensemble des itemsets fréquents vis à vis de ce jeu de données \mathcal{T} est l'ensemble $\mathbf{F}_{\mathcal{T},3} = \{ (f :4), (c :4), (a :3), (b :3), (m :3), (p :3), (fc :3), (fa :3), (fm :3), (cm :3), (cp :3), (ca :3), (am :3), (fca :3), (fcm :3), (fam :3), (cam :3), (fcam :3) \}$.

2.1.3 Complexité et Propriétés

La complexité du problème vient d'une part du nombre d'itemsets à considérer en fonction du nombre d'objets et d'autre part de nombre de transactions dans la base. En effet, soit n objets fréquents dans la base, il y a alors 2^n itemsets possibles. D'autre part, le calcul du support d'un itemset se fait au travers de l'ensemble des transactions. L'efficacité d'une méthode à résoudre ce problème se fera donc par sa capacité à explorer intelligemment l'espace des 2^n itemsets et par son efficacité à calculer le support d'un itemset vis à vis de $|\mathcal{T}|$.

Différents théorèmes et propriétés issus de l'étude de ce problème sont utilisés dans les méthodes proposées pour le résoudre. Nous présentons les propriétés liées à la monotonie du support d'un ensemble et renvoyons le lecteur vers [2] et [3] pour plus de détails.

Monotonie du support. Soit une base de transactions \mathcal{T} sur \mathcal{I} et soient $X, Y \subseteq \mathcal{I}$ deux itemsets. Alors,

$$X \subseteq Y \Rightarrow \mathbf{sup}_{\mathcal{T}}(Y) \leq \mathbf{sup}_{\mathcal{T}}(X)$$

Cette propriété nous permet de dire que si un k-itemset X (i.e. un itemset comprenant k objets) est fréquent, alors l'ensemble Y des k-1-itemsets $\subset X$ est fréquent. Par exemple, (fca :3) est fréquent, de même que (fc :3), (fa :3) et (ca :3). Nous pouvons de manière duale dire que si un k-itemset X est non-fréquent, alors aucun des k+1-itemset Y tel que $X \subset Y$ n'est fréquent. Ces deux observations sont à la base des différents sens de parcours de l'espace de recherche des 2^n itemsets possibles dans la plupart des algorithmes.

2.2 Problème des itemsets clos maximaux

Afin de réduire l'espace de recherche, il a été proposé de contraindre la recherche à l'ensemble des itemsets clos. Un k-itemset X fréquent est clos si $\mathbf{sup}_{\mathcal{T}}(X) > \mathbf{sup}_{\mathcal{T}}(Y) \forall Y$ tel que $X \subset Y$. Un itemset clos est maximal si aucun de ses surensembles n'est fréquent. Ainsi dans notre exemple, l'ensemble des itemsets clos est $\mathbf{C}_{\mathcal{T},3} = \{ (f :4), (c :4), (b :3), (cp :3), (fcam :3) \}$ et l'ensemble des itemsets clos maximaux est $\mathbf{MC}_{\mathcal{T},3} = \mathbf{C}_{\mathcal{T},3} - \{ (f :4), (c :4) \}$. La figure suivante représente le treillis complet des différents itemsets possibles sur les objets fréquents. Y figurent d'une part les itemsets fréquents en vert, d'autre part les itemsets clos en jaune et enfin les itemsets clos maximaux en rouge. Pour plus de lisibilité, les itemsets non-fréquents ne figurent pas dans le treillis.

Zaki et al. et Pas. et al. prouvent dans [4] et [5] que l'intégralité des itemsets fréquents peut être dérivés de l'ensemble des itemsets clos maximaux. Ils proposent alors d'adapter les méthodes existantes pour le calcul des itemsets clos maximaux.

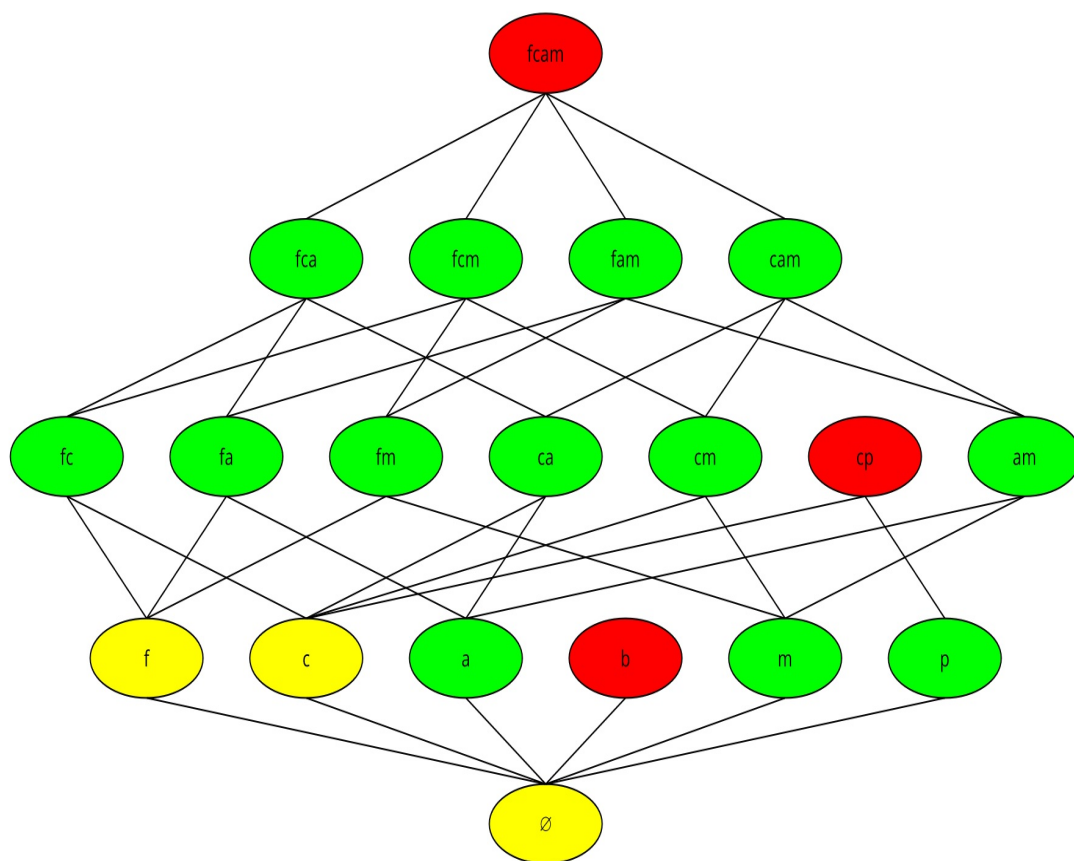


FIGURE 2.1 – Treillis des itemsets fréquents, clos et clos maximaux de notre exemple

Chapitre 3

Etat de l'art

Dans ce chapitre, nous nous attacherons à présenter les principales méthodes de la littérature pour le calcul des itemsets fréquents. La première section sera dédiée à la présentation des méthodes exactes. Nous y aborderons les trois principaux algorithmes que sont A Priori [6], Eclat [7] et FP-Growth [8]. D'autres méthodes dédiées au calcul des itemsets clos maximaux seront également abordées. Enfin nous introduirons quelques méthodes présentant un intérêt pour la problématique du passage à l'échelle. La section 2 sera consacrée aux méthodes métaheuristiques de la littérature. [TO COMPLETE]

3.1 Méthodes exactes

3.1.1 A Priori

Un an après avoir introduit le problème des règles d'associations (et par conséquent le problème de calcul des itemsets fréquents) dans [1], les auteurs ont proposé l'algorithme A Priori [6] restant la référence littéraire du domaine. De nombreux papiers proposent des optimisations et dérivation d'A Priori. Son principe général est la génération de $k+1$ -itemsets candidats à partir des k -itemsets fréquents précédemment calculés. L'algorithme 1 présente la phase de calcul des itemsets fréquents par A Priori. Nous ne présenterons pas l'algorithme calculant les règles d'associations dans cette étude.

Algorithm 1: APriori Itemset Mining

Input : $\mathcal{T}, \text{minsup}$
Output: $\mathbf{F}_{\mathcal{T}, \text{minsup}}$

```
1 begin
2    $C_1 = \{ \{ i \} \mid i \in \mathcal{I} \}$ 
3    $k = 1$ 
4   while  $C_k \neq \{\}$  do
5     // Calcul du support de chaque itemset candidat
6     forall  $t = ( \text{tid}, I ) \in \mathcal{T}$  do
7       forall itemset candidat  $X \in C_k$  do
8         if  $X \subseteq I$  then
9            $X.\text{support}++$ 
10        end
11      end
12    end
13    // Extraction de l'ensemble des itemsets fréquents
14     $\mathbf{F}_k = \{ X \mid X.\text{support} \geq \text{minsup} \}$ 
15    // Génération des candidats de taille k+1
16    forall  $X, Y \in \mathbf{F}_k \mid X[i] = Y[i] \forall i \in [1, k-1] \& X[k] < Y[k]$  do
17       $I = X \cup \{ Y[k] \}$ 
18      if  $\forall J \subset I, |J| = k \text{ et } J \in \mathbf{F}_k$  then
19         $C_{k+1} = C_{k+1} \cup I$ 
20      end
21    end
22     $k++$ 
23  end
24 end
```

[9] [10] [11] [12]

Bibliographie

- [1] R. Agrawal, T. Imielinski, and A.N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–213. ACM Press, 1993.
- [2] F. Bodon. A survey on frequent itemset mining. 2006.
- [3] B. Goethals. Survey on frequent pattern mining.
- [4] M.J. Zaki and C. Hsiao. Charm : An efficient algorithm for closed association rule mining. Technical report, Rensselaer Polytechnic Institute, 1999.
- [5] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *roc. 7th Int. Conf. Database Theory (ICDT'99)*, pages 398–416, Jerusalem, Israel, January 1999.
- [6] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB '94 : Proceedings of the 20th International Conference on Very Large Databases*, pages 487–499. Morgan Kaufmann Publishers Inc., 1994.
- [7] Mohammed Javeed Zaki. Scalable algorithms for association mining. *IEEE Trans. Knowl. Data Eng.*, 12(3) :372–390, 2000.
- [8] J. Han, P. Jian, Y. Yiwen, and M. Runying. *Mining Frequent Patterns without Candidate Generation : A Frequent-Pattern Tree Approach*, volume 8, pages 53–87. 2004.
- [9] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, volume 26(2) of *SIGMOD Record*, pages 255–264. ACM Press, 1997.
- [10] J. Pei, J. Han, and R. Mao. Closet : An efficient algorithm for mining frequent closed itemsets. In *Proc. 2000 ACM-SIGMOD Int. Workshop Data Mining and Knowledge Discovery (DMKD2000)*, pages 11–20, Dallas, TX, 2000.
- [11] C. Borglet. Efficient implementations of a priori and eclat. In *Proceedings of the 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations (FIMI 2003)*, page 90, 2003.
- [12] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D. Yang. H-mine : Hyperstructure mining of frequent patterns in large databases. In *Proc. 2001 Int. Conf. Data Mining (ICDM2001)*, pages 441–448, San Jose, CA, 2001.