Heuristique pour l'extraction de motifs ensemblistes bruités

Karima Mouhoubi, Lucas Létocart et Céline Rouveirol

LIPN, UMR CNRS 7030, Université Paris 13, 99 av. J.B. Clément, 93430 Villetaneuse, France
nom.prénom@lipn.univ-paris13.fr

Résumé. La recherche de motifs ensemblistes dans des matrices de données booléennes est une problématique importante dans un processus d'extraction de connaissances. Elle consiste à rechercher tous les rectangles de 1 dans une matrice de données à valeurs dans {0,1} dans lesquelles l'ordre des lignes et colonnes n'est pas important. Plusieurs algorithmes ont été développés pour répondre à ce problème, mais s'adaptent difficilement à des données réelles susceptibles de contenir du bruit. Un des effets du bruit est de pulvériser un motif pertinent en un ensemble de sous-motifs recouvrants et peu pertinents, entraînant une explosion du nombre de motifs résultats. Dans le cadre de ce travail, nous proposons une nouvelle approche heuristique basée sur les algorithmes de graphes pour la recherche de motifs ensemblistes dans des contextes binaires bruités. Pour évaluer notre approche, différents tests ont été réalisés sur des données synthétiques et des données réelles issues d'applications bioinformatiques.

1 Introduction

La recherche de motifs ensemblistes dans des données booléennes consiste à rechercher tous les rectangles de 1 dans une matrice à valeurs dans $\{0, 1\}$ dans laquelle l'ordre des lignes et colonnes n'est pas important. Lorsque les données booléennes sont le résultat de traitements sur des données numériques issues de processus expérimentaux complexes, celles-ci peuvent alors contenir du bruit. L'effet du bruit va être de fractionner des motifs importants vérifiant certaines contraintes, telle que le support minimal, en un nombre exponentiel de petits fragments non pertinents. La figure 1 illustre un exemple d'un contexte booléen non bruité (matrice A) où, pour un support minimal de 0.3, deux motifs fréquents maximaux peuvent être extraits ainsi que la même matrice mais en introduisant du bruit (matrice B).

La prise en compte du bruit pour la découverte de motifs a fait l'objet d'un nombre important de travaux de recherche tels que Mannila et Seppanen (2004), Besson et al. (2006) et Liu et al. (2006). Pour résoudre ce problème, la plupart des travaux ont repris le principe de recherche par niveau de l'algorithme Apriori d'Agrawal et al. (1993) et sont donc limités à l'utilisation de contraintes anti-monotones pour élaguer l'espace de recherche.

Dans le travail de Mannila et Seppanen (2004), les auteurs recherchent toutes les régions de support minimal σ et qui dépassent un seuil de densité $\delta \in [0, 1]$. Cette approche permet d'extraire toutes les régions verifiant les contraintes du support et de densité. cependant le choix de ces paramètres reste une tâche difficile et nécessite une connaissance préalable sur les données. De plus, la méthode reste très coûteuse puisqu'elle utilise une recherche par niveau.

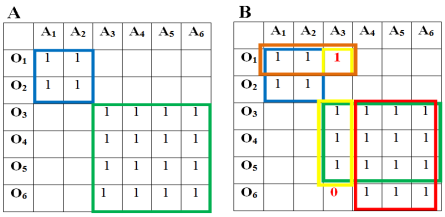

FIG. 1 – *Effet du bruit sur le nombre de motifs maximaux dans une matrice booléenne.*

Besson et al. (2006) se sont intéressés à la recherche des ensembles d'objets et d'attributs qui sont fortement associés, appelés *bi-ensembles*. Pour cela, ils utilisent des contraintes de densité sur chaque ligne et colonne. En effet, l'utilisateur doit fixer deux paramètres α et β de telle sorte que les bi-ensembles extraits ne doivent pas contenir plus de α zéros sur chaque ligne et β zéros sur chaque colonne. De plus, les bi-ensembles doivent être maximaux et vérifier une contrainte de pertinence qui exige que le nombre d'exceptions sur chaque ligne (resp. colonne) d'un bi-ensemble soit inférieur à celui de toute autre ligne (resp. colonne) du reste de la matrice de données, avec le paramètre γ défini par l'utilisateur. Citons également le modèle *Approximate Frequent Itemsets* proposé dans Liu et al. (2006) qui tolère aussi, comme dans Besson et al. (2006), une fraction contrôlée d'exceptions sur chaque ligne ainsi que sur chaque colonne. Les auteurs ont utilisé le principe de recherche par niveau et ont proposé un support qui prend en compte la longueur des motifs ainsi que les taux d'erreur tolérés sur chaque ligne et colonne. Ces modèles permettent de récupérer des sous-matrices intéressantes mais restent très coûteux en temps d'exécution car les traitements se font sur chaque ligne et colonne.

L'objectif de ce travail est donc d'explorer une voie alternative pour résoudre ce problème de manière efficace : utiliser et adapter des algorithmes d'optimisation combinatoire, notamment des algorithmes de graphes et de les combiner avec des méthodes de la fouille de données. L'approche que nous proposons consiste à construire à partir d'une matrice booléenne des graphes bipartis pondérés puis à rechercher les sous-graphes denses les plus grands dans ces derniers en se basant sur les algorithmes de flot maximal/coupe minimale. Pour évaluer cette approche, nous l'avons implémentée et testée sur des jeux de données synthétiques ainsi que sur des données réelles. Nous présentons dans le cadre de ce travail les résultats obtenus sur des données biologiques. Cependant, l'approche peut également s'appliquer et s'adapter à d'autres types de données comme l'extraction de communautés dans des réseaux sociaux.

2 Préliminaires

Définition 1 (Contexte formel et motif)
Soit O un ensemble fini d'objets, A un ensemble fini d'attributs, et R une relation binaire entre ces deux ensembles. On appelle contexte formel le triplet $D = (O, A, R)$ qui peut être modélisé par une matrice booléenne où les lignes et les colonnes correspondent respectivement aux objets et aux attributs. Un motif *m* est un sous-ensemble d'attributs de *A*.

Définition 2 (Motif fréquent)
Un objet $o \in O$ supporte un motif *m* si tous les attributs de *m* appartiennent à la description de *o*.

Le support d'un motif est le rapport entre le cardinal de l'ensemble des objets qui le contiennent et le cardinal de l'ensemble des objets du contexte. Un motif est dit fréquent relativement à un support minimal *minsup* $\in [0, 1]$ si son support est supérieur ou égal à *minsup*. Un motif fréquent est dit maximal s'il n'est sous-motif d'aucun des motifs fréquents du contexte.

Définition 3 (Graphe biparti orienté et densité)
Un graphe biparti orienté G est défini par deux ensembles de sommets V_1 et V_2 et un ensemble d'arcs E telle que chaque arc ait une extrémité dans V_1 et l'autre dans V_2. Dans le cadre de ce travail, nous définissons la densité d'un graphe biparti G par le rapport $\frac{|E|}{|V_1|\times|V_2|}$. Un graphe est dit dense, relativement à $\delta \in [0, 1]$, si sa densité est supérieure ou égale à δ.

Définition 4 (Sommet fortement associé à un ensemble de sommets)
Cette propriété permet de déterminer dans un graphe biparti $G = (V_1, V_2, E)$ l'ensemble des sommets de V_1 ayant un degré important et/ou reliés à des sommets de V_2 de degrés élevés. Un sommet $v_i \in V_1$ de degré $d(v_i)$ est fortement associé aux sommets de V_2 si et seulement si $\sum_{v_j \in V_2 \wedge d(v_j)\neq 0} (\frac{d(v_i)}{max_{v_k \in V_1}(d(v_k))} + \frac{d(v_j)}{max_{v_k \in V_2}(d(v_k))}) > max_{v_k \in V_1}(d(v_k))$.

Définition 5 (Coupe minimale)
Soit $G = (V, E)$ un graphe orienté possédant un sommet "source" s de degré sortant non nul et un sommet "destination" t de degré entrant non nul. À tout arc (x, y) est associé un entier $c(x, y)$ positif ou nul, sa capacité. Une coupe est une partition de V en $S \cup T$ où $s \in S$ et $t \in T$. La capacité de la coupe notée $c(S, T)$ est la somme des capacités des arcs de S vers T. Une coupe est dite minimale si sa capacité est minimale.

3 Recherche de régions denses

Nous présentons dans cette section notre méthodologie pour l'extraction de motifs dans des contextes bruités. Notre objectif est de rechercher les régions denses en 1 maximales dans des matrices de données booléennes. Partant d'un motif m_0 de densité 1 (rectangle de 1), notre algorithme permet d'extraire toutes les sous-matrices denses maximales d qui incluent m_0 et qui respectent les contraintes suivantes : 1) chaque colonne de d a une densité supérieure à un seuil δ, et 2) chaque ligne de d est fortement associée à ses colonnes (Définition 4). De cette manière, nous arrivons à extraire toutes les régions maximales qui incluent m_0 et possédant une densité supérieure à δ. Notons que chaque sous-matrice extraite est maximale du fait qu'aucune de ses sur-matrices ne vérifie les contraintes 1 et 2.

Notre objectif étant d'extraire des régions denses maximales, nous avons opté pour les motifs maximaux comme motifs initiaux que nous calculons à l'aide de l'implantation de Borgelt d'Apriori (Borgelt et Kruse (2002)). Partant d'un motif maximal m_0, nous construisons le graphe correspondant (Algorithme 2) puis nous calculons une coupe minimale. Pour celà, nous avons opté pour l'algorithme de Cherkassky et Goldberg (1997). Les capacités affectées aux arcs sont donc adaptées de manière à récupérer, après le calcul de la coupe minimale, un sous-graphe dense qui comporte en plus des sommets attributs de m_0 un ensemble de lignes l_0 qui sont fortement associées à ces attributs. Lors de la prochaine étape, nous construisons le graphe correspondant aux lignes l_0 de manière à récupérer, après le calcul de la coupe minimale, un sous-ensemble d'attributs ayant des densités supérieures à δ pour ces lignes l_0. Comme illustré dans l'algorithme 1, ce processus est répété jusqu'à ce que le sous-graphe dense extrait à l'étape n soit identique à celui extrait à l'étape $n - 1$, dans ce cas notre sous-graphe ne peut

plus être augmenté et le processus est arrêté. Comme illustré dans l'algorithme 2, la construction d'un graphe lors de l'appel avec les observations diffère de celle avec les attributs. Cette différence réside dans l'affectation des poids des arcs puisque les critères de sélection d'une observation sont différents de ceux d'un attribut tenant compte du fait que le nombre d'attributs dans les matrices des données est beaucoup plus grand que le nombre d'observations (matrices d'expression de gènes). De plus, nous nous intéressons aux cas où les observations ne sont pas très denses mais dans lesquelles figurent des attributs très denses.

Algorithme 1 : L'algorithme de recherche des régions denses

Entrées : D : matrice des données, M_max : motifs maximaux, δ : densité minimale de chaque colonne(attribut)
Sorties : SGD : l'ensemble des sous-graphes denses
début
  pour tous les $m_i \in M\_max$ **faire**
    C_{pre}={} ; L_{pre}={} ; i=0
    G_i=CONSTRUIRE_GRAPHE(m_i, D);
    (L_i, C_i)= S_COUPE-MINIMALE(G_i);
    tant que ($C_{pre} \neq C_i$ *ou* $L_{pre} \neq L_i$) **faire**
      SGD=SGD $\cup$ (L_i, C_i) ;
      C_{pre}=C_i ; L_{pre}=L_i ; i++;
      si (*i est impair*) **alors**
        G_i=CONSTRUIRE_GRAPHE(L_{pre}, D, δ);
      sinon
        G_i=CONSTRUIRE_GRAPHE(C_{pre}, D);
      (L_i, C_i)= S_COUPE-MINIMALE(G_i);
fin

Algorithme 2 : Construction des Graphes

Entrées : A : ensemble des sommets, D : matrice des données, δ : paramètre de densité
Sorties : $G(V, E)$: le graphe construit
début
  $V = A \cup \{s, t\}$;
  pour tous les $a_i \in A$ **faire**
    E=$E \cup (s, a_i)$; poids(s, a_i)= $+\infty$; /* poids (s, a_i): le poids de l'arc (s, a_i) */
  suivant *les éléments de* A **faire**
    cas où *(les éléments de A sont des lignes)*
    pour tous les $a_i \in A$ **faire**
      pour tous les $D[a_i][a_j] = 1$ **faire**
        V=$V \cup a_j$; E=$E \cup (a_i, a_j)$; poids(a_i, a_j)= $\frac{100}{|A|}$;
    pour tous les $a_j \in V \backslash (A \cup \{s, t\})$ **faire**
      E=$E \cup (a_j, t)$;
      poids (a_j, t)= 2x(100xδ) - $poids^-(a_j)$; /* $poids^-(a_j)$: la somme des poids des arcs entrants vers a_j */
    cas où *(les éléments de A sont des colonnes)*
    pour tous les $a_i \in A$ **faire**
      pour tous les $D[a_j][a_i] = 1$ **faire**
        V=$V \cup a_j$; E=$E \cup (a_i, a_j)$;
        poids(a_i, a_j)= $\left(\frac{d^+(a_i)}{max_{a_k \in A}(d^+(a_k))} + \frac{d^-(a_j)}{max_{a_k \in A_j}(d^-(a_k))}\right)$ x $\frac{100}{|A|}$
    pour tous les $a_j \in V \setminus (A \cup \{s, t\})$ **faire**
      E=$E \cup (a_j, t)$; poids (a_j, t)= $max_{a_k \in A_j}(d^-(a_k))$x $\frac{200}{|A|}$ - $poids^-(a_j)$
fin

4 Expérimentations et résultats

Nous avons implémenté l'approche avec le langage C et utilisé pour l'expérimentation un ordinateur équipé d'un microprocesseur intel(R) Pentium(R) 4 (3 GHz) et d'une mémoire vive

de 2GB. L'évaluation a été effectuée sur des données synthétiques et réelles et nos résultats ont été comparés à ceux obtenus par l'algorithme $Dense$ de Mannila et Seppanen (2004).

Afin d'étudier la pertinence de l'approche, nous avons construit des jeux de données dans lesquels nous avons introduit des régions denses et nous avons comparé les résultats de l'algorithme à ce qui devait être extrait (les régions denses introduites). Les régions denses peuvent être soit disjointes (n'ayant aucune ligne ou colonne en commun) ou se recouvrir (ayant des lignes et/ou colonnes en commun). Nous avons fait varier les tailles des régions denses ainsi que leurs taux de recouvrement. Notre algorithme assure l'extraction de toutes les régions denses disjointes et maximales vérifiant la contrainte de densité minimale indépendamment. Pour deux régions recouvrantes, s'il y a fort recouvrement des régions par rapport à δ, l'approche extrait une seule région, union des deux régions. Dans le cas contraire, elle extrait bien les régions indépendamment. Pour vérifier la robustesse de l'approche, nous l'avons testé sur des données dans lesquelles nous introduisons un bruit aléatoire et nous avons comparé nos résultats à ceux obtenus par $Dense$ avec les mêmes paramètres de support et de densité. Nous avons conclu que le nombre de résultats extraits par $Dense$ reste très élevé par rapport à notre algorithme ce qui est dû au fait que $Dense$ renvoie tous les sous-ensembles d'un motif vérifiant les contraintes de densité et de support minimal. De plus, notre approche assure l'extraction des régions denses introduites même avec un taux de bruit élevé (20%).

Nous avons aussi évalué l'approche sur des données réelles d'expression de gènes en faisant varier le seuil de densité δ, pour un support minimal de 0.2. Le tableau 2 montre les résultats obtenus par notre approche ainsi que par l'algorithme $Dense$ sur les données de Spellman et al. (1998). Elles se composent d'une série de 69 puces à ADN mesurant l'expression de 407 gènes pendant le cycle cellulaire chez la levure. Leur densité est de 0.27 (Elati et al. (2007)) et pour un support minimal de 0.2, le nombre de maximaux fréquents calculés est 734.

	Notre algorithme				Dense		
δ	nombre de résultats	densités (min-max-moy)	Taille max	Temps	nombre de résultats	Taille max	Temps
0.5	5380 - 3889 - 229	0.56 - 1 - 0.81	95	1m 32s	-	3	58m
0.6	5096 - 3605 - 363	0.66 - 1 - 0.85	90	1m 30s	-	3	43m
0.7	3485 - 2808 - 555	0.75 - 1 - 0.89	43	1m 23s	-	3	874m
0.8	1208 - 1140 - 714	0.82 - 1 - 0.96	13	27s	74356006	21	538m

TAB. 1 – *Résultats des expérimentations sur les données de Spellman et al. (1998)*

Nous calculons pour chaque algorithme le nombre de motifs extraits, leurs longueurs maximales ainsi que le temps d'exécution en minutes (m) et secondes (s). Nous présentons dans la deuxième colonne respectivement le nombre de résultats total extraits vérifiant la contrainte de densité minimale, le nombre de résultats différents ainsi que de maximaux différents. La troisième colonne contient les densités des résultats obtenus (densité minimale, maximale et moyenne). Notons que le nombre de régions denses maximales reste égal au nombre de maximaux initiaux, 734. Les résultats obtenus qui sont résumés dans le tableau1 nous montrent que notre algorithme permet d'extraire des motifs de densités importantes, de grandes tailles et en un temps d'exécution raisonnable par rapport à $Dense$. Comme on le voit, les expérimentations lancées avec l'algorithme $Dense$ pour des densités inférieures à 0.8 n'ont pu être terminées par cause d'un manque d'espace mémoire ou temps d'exécution limite. Cependant,

les résultats obtenus par notre approche ne sont pas tous différents les uns des autres (colonne 2). En effet, le bruit pulvérise un maximal fréquent en plusieurs maximaux ; de ce fait, plusieurs motifs initiaux peuvent mener à une seule région dense ce qui explique les résultats possiblement redondants.

5 Conclusion

Nous avons présenté une nouvelle approche basée sur les algorithmes de graphes pour la recherche de motifs dans des contextes bruités. Les résultats sont très encourageants concernant la qualité et la taille des motifs extraits et en un temps d'exécution raisonnable. En guise de perspectives, nous envisageons, dans un premier temps, de proposer et d'explorer de nouvelles stratégies pour extraire toutes les régions maximales et denses sans calculer tous les maximaux fréquents pour remédier au problème des résultats redondants. Au delà de cette amélioration, nous comptons adapter notre approche pour permettre l'extraction de motifs de natures différentes (séquence, arbre) dans des environnements bruités.

Références

Agrawal, R., T. Imielinski et A. Swami (1993). Mining Association Rules between sets of Items in Large Databases, *proc. ICDM'93*, pp 207-213.

Besson, J., J. F. Boulicaut et C. Robardet (2006). Mining a New Fault- Tolerant Pattern Type as an Alternative to Formal Concept Discovery. *LNCS*, 4068 :144-157.

Borgelt, C. et R. Kruse. (2002). Induction of Association Rules : Apriori Implementation. *15th Conference on Computational Statistics*, 395-400.

Cherkassky, B. V. et A. V. Goldberg (1997). On implementing the pushrelabel method for the maximum flow problem. *Algorithmica ISSN*, 19(4) :390-410.

Elati, M., P. Neuvial, M. Bolotin-Fukuhara et al. (2007). LICORN : learning co-operative regulation networks from expression data. *In Bioinformatics*, 23 :2407-2414.

Liu, J., S. Paulsen, X. Sun et al. (2006). Mining Approximate Frequent Itemsets In the Presence of Noise : Algorithm and Analysis. *SIAM*.

Mannila, H. et J. K. Seppanen (2004). Dense Itemsets. *In Proceedings of the 10th ACM SIGKDD. Int. conf. on knowledge discovery and data mining*, 683-688.

Spellman, P.T., G. Sherlock et al. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*.

Summary

Itemset mining in boolean matrices is an important step in a knowledge extraction process. It consists in finding all rectangles of 1 in a boolean matrix in which the order of the rows and columns is not important. Several algorithms have been developed to address this problem, but it is difficult to adapt classical itemset mining algorithms to real data that may contain noise. One effect of noise is to shatter relevant itemsets into a set of small irrelevant itemsets, yielding an explosion in the number of resulting itemsets. In this work, we propose a new heuristic approach based on a graph algorithm for the efficient extraction of itemset patterns in noisy binary contexts. To evaluate our approach, various tests have been performed on both synthetic data and real datasets from bioinformatic applications.