Extraction de Liens Fréquents dans les Réseaux Sociaux

Erick Stattner*, Martine Collard*

* Laboratoire LAMIA
Université des Antilles et de la Guyane, France
{estattne, mcollard}@univ-ag.fr

Résumé. Cet article présente *FLMin*, une nouvelle méthode d'extraction de motifs fréquents dans les réseaux sociaux. Contrairement aux méthodes traditionnelles qui s'intéressent uniquement aux régularités structurelles, l'originalité de notre approche réside dans sa capacité à exploiter la structure et les attributs des noeuds pour extraire des régularités, que nous appelons "*liens fréquents*", dans les liens entre des noeuds partageant des caractéristiques communes.

1 Introduction

Les approches traditionnelles de la fouille de données reposent sur l'hypothèse implicite selon laquelle les données sont indépendantes et identiquement distribuées (*IID*). Si cette restriction s'avère être cohérente au regard du problème classique d'inférence statistique, elle ignore toutefois les relations de dépendance et de corrélation, inhérentes à de nombreux phénomènes du monde réel, qui émergent fréquemment d'interactions complexes entre des entités.

Ces dernières décennies ont ainsi vu naitre la "science des réseaux" (Barabasi, 2002), une discipline qui vise à l'étude des relations entre des entités. Récemment, le domaine du *"link mining"* (Getoor et Diehl, 2005) a tenté d'appliquer les concepts du data mining aux réseaux, en s'intéressant entre autres à l'extraction de motifs. Dans ce domaine, les travaux se sont limités à l'exploitation de la structure du réseau, à travers la recherche de sous-graphes fréquents.

Cet article présente une approche nouvelle du problème de la recherche de motifs fréquents dans les réseaux sociaux. L'originalité de notre solution est qu'elle combine à la fois la structure et les attributs des noeuds, pour extraire des motifs réguliers, appelés *"liens fréquents"*, au sein des liens qui connectent des noeuds possédant des caractéristiques communes.

Ce papier est organisé en cinq sections. La Section 2 passe en revue les travaux sur l'extraction de motifs dans les réseaux et définit le problème de la recherche des liens fréquents. La Section 3 détaille et discute *FLMin*, la méthode proposée. Dans la Section 4, nous évaluons les performances de notre solution. Enfin, la Section 5 conclut et présente nos travaux futurs.

2 Contexte

2.1 État de l'art

Dans le domaine de la modélisation réseau, la définition la plus courante et la plus largement admise d'un motif est celle du "sous-graphe" (Inokuchi et al., 2000; Kuramochi et

Karypis, 2005). L'approche traditionnelle consiste en l'utilisation d'étiquettes associées aux noeuds et aux liens. En utilisant une telle représentation, le problème est ainsi ramené à la recherche d'étiquettes qui se retrouvent fréquemment.

Deux approches peuvent être distinguées. La première regroupe les méthodes qui s'intéressent aux sous-graphes dans des bases de données de réseaux (Inokuchi et al., 2000) et la seconde concerne les méthodes qui effectuent la recherche au sein d'un même réseau (Kuramochi et Karypis, 2005). Quelque soit l'approche considérée, la grande majorité des techniques exploite les propriétés de l'algorithme fondateur *Apriori* (Agrawal et Srikant, 1994), à travers un processus d'extraction en deux phases : (i) une étape de génération des sous-graphes candidats et (ii) une étape d'évaluation qui mesure la fréquence des candidats en s'appuyant sur les propriétés de l'isomorphisme. Par exemple, Inokuchi et al. (Inokuchi et al., 2000) ont présenté *AGM*, un algorithme basé sur *Apriori*, pour extraire les sous-graphes dans un ensemble de réseaux. Kuramochi et Karypis (Kuramochi et Karypis, 2001) ont proposé l'algorithme *FSG*, qui utilise une représentation minimaliste pour réduire le stockage et le temps de calcul.

2.2 Concept de "liens fréquents"

Contrairement aux travaux dans ce domaine qui s'intéressent uniquement aux régularités structurelles, nous proposons ici une nouvelle approche du problème, qui consiste à rechercher des régularités au sein des attributs associés aux noeuds connectés du réseau.

Plus formellement, soit $G = (V, E)$ un réseau. V est l'ensemble des noeuds et E l'ensemble des liens avec $E \subseteq V \times V$. V est défini comme une relation $R(A_1, ..., A_p)$ ou chaque A_i est un attribut. Ainsi, chaque noeud $v \in V$ est défini par le tuple $(a_1, ..., a_p)$ où $\forall k \in [1..p], v[A_k] = a_k$, la valeur de l'attribut A_k dans v. Un item est une expression logique $A = x$ où A est un attribut et x une valeur. Un itemset est une conjonction d'items, par exemple $A_1 = x$ *et* $A_2 = y$ *et* $A_3 = z$. Posons m_1 et m_2 deux itemsets et V_{m_1}, V_{m_2}, respectivement les ensembles de noeuds dans V qui satisfont m_1 et m_2. Nous notons $E_{(m_1,m_2)}$ l'ensemble des liens connectant des noeuds de V_{m_1} à des noeuds de V_{m_2}, i.e.
$E_{(m_1,m_2)} = \{e \in E \; ; \; e = (a, b) \quad a \in V_{m_1} \;\; et \;\; b \in V_{m_2}\}$

Définition 1. Nous appelons *support* de $E_{(m_1,m_2)}$, le pourcentage de liens appartenant à $E_{(m_1,m_2)}$, i.e. $supp(E_{(m_1,m_2)}) = \frac{|E_{(m_1,m_2)}|}{|E|}$

Définition 2. Nous disons qu'il y a un lien fréquent entre m_1 et m_2, et nous notons (m_1, m_2), si le support de $E_{(m_1,m_2)}$ est plus grand qu'un seuil de support minimum β, i.e. $supp(E_{(m_1,m_2)}) > \beta$

Définition 3. Soit I l'ensemble des itemsets dans V, nous définissons FL l'ensemble des liens fréquents comme, $FL = \cup_{m_1 \in I, m_2 \in I} \{E_{(m_1,m_2)} \; ; \; supp(E_{(m_1,m_2)}) > \beta\}$

Découvrir tous les liens fréquents dans un réseau peut-être très couteux si l'espace de recherche est grand. Cependant, comme l'illustre la figure 1, les liens fréquents peuvent être observés dans deux types de configuration : (a) m_1 et m_2 sont tous les deux fréquents ou (b) au moins un des deux est très fréquent. Notre approche consiste à explorer l'espace de recherche réduit aux liens impliquant un itemset fréquent, en posant un seuil de fréquence minimum α. Dans le contexte des réseaux, les attributs des noeuds sont généralement peu nombreux, ce qui rend le parcours du treillis des itemsets fréquents moins couteux.

Nous pouvons distinguer trois cas : (i) $V_{m_1} = V_{m_2}$, (ii) $V_{m_1} \cap V_{m_2} \neq \emptyset$ et (iii) $V_{m_1} \cap V_{m_2} = \emptyset$. Le cas (i) renvoie au problème de la recherche de sous-graphes et le cas (ii) peut-

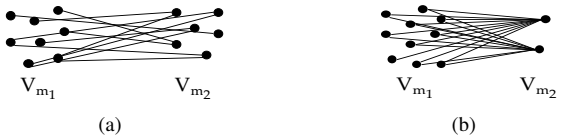


FIG. 1 – *Configurations d'apparition des liens fréquents*

être résolu à partir du (iii). Dans ce travail, nous nous intéressons donc au troisième cas : $V_{m_1} \cap V_{m_2} = \emptyset$. Dans ce contexte, si le lien (m_1, m_2) est fréquent, $|V_{m_1}| \times |V_{m_2}| > \beta \times |E|$ puisque $|V_{m_1}| \times |V_{m_2}| > |E_{(m_1,m_2)}|$. L'espace de recherche peut ainsi être réduit aux paires d'ensembles disjoints V_{m_1} et V_{m_2} qui satisfont la propriété (a) suivante : $|V_{m_2}| > \frac{\beta \times |E|}{|V_{m_1}|}$.

3 Extraction des Liens Fréquents

Rechercher les liens fréquents peut-être difficile car le nombre de liens joue un rôle clé dans les phases de calcul. Ainsi, plutôt que de parcourir l'ensemble des liens, la recherche des liens fréquents s'effectue en deux étapes comme décrit sur l'algorithme FLMin (algorithme 1).

Nous commençons par filtrer les noeuds pour extraire les itemsets fréquents au delà d'un seuil minimum α, conformément à la remarque faite à la section 2.2. Nous utilisons pour cela un algorithme d'extraction pour les données non-binaires (Han et al., 2007) (voir ligne 1 algorithme 1). Ici, l'optimisation de la génération des itemsets n'est pas un point sensible puisque l'espace de recherche n'est pas aussi large que dans les cas typiques de data mining, en raison du peu d'attributs que présentent généralement les noeuds dans les réseaux sociaux.

La seconde phase de *FLMin* consiste à évaluer la fréquence des paires d'itemsets qui vérifient l'équation (a) (voir lignes 6 à 15 de l'algorithme). Nous avons précédemment défini qu'une paire d'itemsets est un lien fréquent si :

$$\frac{|\{e \in E\ ;\ e = (a,b) \quad a \in V_{m_1}\ et\ b \in V_{m_2}\}|}{|E|} > \beta \tag{1}$$

Une propriété intéressante de l'inéquation 1, dans le cas des réseaux non-orientés, est que si (m_1, m_2) est fréquent, le lien (m_2, m_1) est également fréquent.

De nombreux travaux se sont intéressés aux mesures d'intérêt pour évaluer et ordonner les informations extraites (Brisson et Collard, 2008). Soient deux seuils de pertinence $\gamma_{\leftarrow}$ et $\gamma_{\rightarrow}$, nous proposons trois mesures d'intérêt pour évaluer la pertinence d'un lien fréquent.

La première est la **mesure de dépendance** $m_1 \dashrightarrow m_2$, qui évalue la proportion de noeuds de V_{m_1} connectés à des noeuds de V_{m_2}. Le lien (m_1, m_2) est pertinent selon $m_1 \dashrightarrow m_2$ si :

$$\frac{|\{e \in E\ ;\ e = (a,b) \quad a \in V_{m_1}\ et\ b \in V_{m_2}\}|}{|\{e \in E\ ;\ e = (a,b) \quad a \in V_{m_1}\}|} > \gamma_{\rightarrow} \tag{2}$$

La **mesure de dépendance** $m_1 \dashleftarrow m_2$ évalue le pourcentage de noeuds dans V_{m_2} qui reçoivent une connexion de noeuds dans V_{m_1}. Un lien est pertinent selon $m_1 \dashleftarrow m_2$ si :

$$\frac{|\{e \in E\ ;\ e = (a,b) \quad a \in V_{m_1}\ et\ b \in V_{m_2}\}|}{|\{e \in E\ ;\ e = (a,b) \quad b \in V_{m_2}\}|} > \gamma_{\leftarrow} \tag{3}$$

Algorithm 1 *FLMin* : Algorithme d'Extraction des Liens Fréquents

Require: $G = (V, E)$, $\alpha \in [0..1]$ et $\beta \in [0..1]$
1. $I \leftarrow$ Générer les itemsets fréquents selon le seuil α
2. $FL \leftarrow \emptyset$
3. **for all** itemset $m_1 \in I$ **do**
4. $\quad V_{m_1} \leftarrow \{v \in V \ ; \ v \in m_1\}$
5. $\quad E_{m_1} \leftarrow \{e \in E \ ; \ e = (a, b), \quad a \in V_{m_1}\}$
6. $\quad$ **for all** itemset $m_2 \in I$ **do**
7. $\quad\quad V_{m_2} \leftarrow \{v \in V \ ; \ v \in m_2\}$
8. $\quad\quad$ **if** $m_1 \neq m_2$ et $V_{m_1} \cap V_{m_2} = \emptyset$ et $|V_{m_2}| > \frac{\beta \times |E|}{|V_{m_1}|}$ **then**
9. $\quad\quad\quad l \leftarrow (m_1, m_2)$
10. $\quad\quad\quad E_{(m1,m2)} \leftarrow \{e \in E_{m_1} \ ; \ e = (a, b) \ \ b \in V_{m_2}\}$
11. $\quad\quad\quad l.support \leftarrow \frac{|E_{(m1,m2)}|}{|E|}$
12. $\quad\quad\quad FL.add(l)$
13. $\quad\quad$ **end if**
14. $\quad$ **end for**
15. **end for**
16. $FL \leftarrow \{l \in FL \ ; \ l.support > \beta\}$
17. **return** FL

Les mesures (2) et (3) peuvent suggérer un lien de causalité qui n'est pas statistiquement vrai. Nous proposons donc un **test de dépendance symétrique** basé sur le principe du *lift* :

$$\frac{|E| \ \times \ |\ \{e \in E \ ; \ e = (a, b), \quad a \in V_{m_1} \ et \ b \in V_{m_2}\}\ |}{|\ \{e \in E \ ; \ e = (a, b) \quad a \in V_{m_1}\}\ | \times |\ \{e \in E \ ; \ e = (a, b) \quad b \in V_{m_2}\}\ |} > 1 \qquad (4)$$

Une caractéristique importante de tout algorithme dont l'objectif est d'analyser des réseaux est la capacité à s'adapter à tous les types de réseau (orientés, non-orientés, multipartites, etc).

FLMin peut-être directement appliqué aux réseaux non-orientés si les liens sont stockés dans les deux directions. Pour les réseaux multipartites, l'algorithme de génération des itemsets ne peut pas être appliqué directement. Une solution consiste à commencer par générer les itemsets pour chaque type de noeud et appliquer ensuite *FLMin*.

Concernant la complexité, les étapes les plus coûteuses sont (a) la génération des ensembles E_{m_1} et $E_{(m_1,m_2)}$ et (b) la phase de combinaisons des (m_1, m_2). Un moyen efficace d'implémenter la tâche (a) consiste à utiliser une structure de noeud qui stocke ses voisins. La génération de E_{m_1} et $E_{(m_1,m_2)}$ peut ainsi s'effectuer en parcourant uniquement les noeuds. Le nombre de combinaisons de la tâche (b) peut-être réduit en commençant par les ensembles les plus grands. En effet, tout sous-item d'un itemset fréquent est également fréquent.

4 Expérimentations

Le jeu de données utilisé dans nos expériences est un réseau obtenu avec *Episims*, un outil de simulation conçu pour reproduire les déplacements d'individus dans la ville de Portland. Dans ce réseau, deux individus sont connectés s'ils ont été géographiquement proches durant la simulation. Les données ont été traitées de façon à ce que chaque individu soit caractérisé

Liens fréquents	Support
((* ;* ;2 ;* ;*),(* ;* ;1 ;* ;*))	0.295
((* ;* ;1 ;* ;*),(* ;* ;2 ;* ;*))	0.295
((* ;* ;1 ;* ;*),(* ;* ;* ;2 ;*))	0.294

(a)

Liens de dépendance	$m_1 \dashrightarrow m_2$
((* ;5 ;1 ;1 ;*),(* ;* ;2 ;* ;*))	0.715
((* ;5 ;1 ;* ;1),(* ;* ;2 ;* ;*))	0.723
((* ;4 ;1 ;1 ;*),(* ;* ;* ;2 ;*))	0.738

(b)

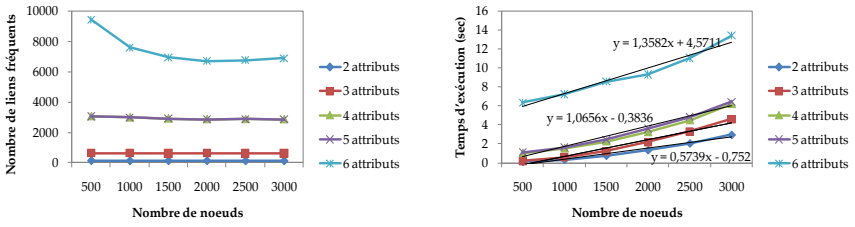
FIG. 2 – *Exemples de (a) liens fréquents et (b) liens de dépendance obtenus pour la configuration* $|V| = 500$, $|R| = 5$ *avec (a)* $\beta = 0.29$ *et (b)* $\beta = 0.2$ *et* $\gamma_{\rightarrow} = 0.7$

par : (1) le numéro de maison (2) la classe d'age, i.e. $\lfloor \frac{age}{10} \rfloor$ (3) le sexe (1-homme, 2-femme) (4) le statut professionnel (1-a une profession, 2-n'en possède pas) (5) le type de relation avec le chef de famille (1-conjoint, partenaire, ou chef de famille, 2-enfant, 3-parent adulte, 4-autre) et (6) la classe de contact, i.e. $\lfloor \frac{degree}{2} \rfloor$.

Les tests ont été réalisés de façon empirique en fixant $\alpha = 0.05$. La taille du réseau est modifiée en prélevant des sous-graphes dans le réseau global. Le nombre d'attributs $|R|$ évolue en supprimant certains attributs des noeuds. *FLMin* a été développé en JAVA et intégré au sein de l'outil graphique *GT-FLMin* [1]. Tous les tests ont été menés sur un Intel Core 2 Duo P8600, Java JDK 1.6 et moyennés sur 100 exécutions.

Dans une approche qualitative, nous comparons sur la figure 2 les motifs obtenus pour la configuration $|V| = 500$ et $|R| = 5$ (attribut 6 supprimé) en utilisant (a) la mesure de fréquence avec $\beta = 0.29$ et (b) la mesure de dépendance $m_1 \dashrightarrow m_2$ avec $\beta = 0.2$ et $\gamma_{\rightarrow} = 0.7$. Le caractère '*' signifie que l'attribut peut prendre n'importe quelle valeur. La première ligne de la table (b) indique que 71.5% des liens du réseau connectent, à une femme, des hommes d'une quarantaine d'années qui ont une profession.

La figure 3 compare, selon la taille du réseau, l'impact du nombre d'attributs sur (a) le nombre de liens fréquents et (b) le temps de calcul avec un seuil d'acceptabilité $\beta = 0.02$.



(a) (b)

FIG. 3 – *Evolution du (a) nombre de liens fréquents et (b) temps d'exécution quand* $\beta = 0.02$

Quelque soit la taille du réseau, le nombre de motifs extraits reste relativement stable pour un nombre d'attribut donné. Ce résultat peut s'expliquer par deux facteurs. Le premier concerne la nature du jeu de données. En effet, de nombreux attributs sont binaires, et donc en s'intéressant à un sous-graphe, la probabilité de retrouver les mêmes itemsets que dans le réseau global est forte. Le second concerne les comportements humains en général. En effet, les facteurs sous-jacents qui créent ou influencent les comportements se retrouvent également

1. *GT-FLMin* : http ://erickstattner.com/GT-FLMin/

à des échelles plus petites. Ainsi, lorsque le sous-ensemble est suffisamment pertinent, la distribution des données est telle, qu'il devient possible d'extraire une grande majorité des motifs.

En ce qui concerne le temps nécessaire à l'extraction (cf Figure 3(b)), nous constatons que le temps de calcul croit linéairement avec la taille du réseau pour un nombre d'attributs donné.

5 Conclusion

Dans cet article, nous nous sommes intéressés à la recherche de motifs fréquents dans les réseaux sociaux. (i) Nous avons présenté et défini la recherche de liens fréquents, un moyen nouveau et original de prendre en compte la structure du réseau et les attributs des noeuds dans la recherche de motifs. (ii) Nous avons proposé *FLMin*, le premier algorithme de recherche des liens fréquents dans les réseaux sociaux. (iii) Une première implémentation de notre solution a été proposée à travers l'outil graphique *GT-FLMin*.

A court terme, nous voulons améliorer les performances de *FLMin* en réduisant sa combinatoire. Notre solution pourrait également extraire des structures plus complexes impliquant plusieurs itemsets qui représenteraient alors des sous-graphes. A long terme, nous envisageons d'utiliser les motifs extraits pour aborder le problème de la prédiction de liens.

Références

Agrawal, R. et R. Srikant (1994). Fast algorithms for mining association rules in large databases. In *20th International Conference on Very Large Data Bases*, pp. 487–499.

Barabasi, A. L. (2002). *Linked : The New Science of Networks*. Perseus Books.

Brisson, L. et M. Collard (2008). How to semantically enhance a data mining process ? In *ICEIS*, pp. 103–116.

Getoor, L. et C. P. Diehl (2005). Link mining : a survey. *SIGKDD Explor. 7*, 3–12.

Han, J., H. Cheng, D. Xin, et X. Yan (2007). Frequent pattern mining : current status and future directions. *Data Min. Knowl. Discov. 15*, 55–86.

Inokuchi, A., T. Washio, et H. Motoda (2000). An apriori-based algorithm for mining frequent substructures from graph data. In *PKDD*, pp. 13–23.

Kuramochi, M. et G. Karypis (2001). Frequent subgraph discovery. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pp. 313–320.

Kuramochi, M. et G. Karypis (2005). Finding frequent patterns in a large sparse graph. *Data Min. Knowl. Discov. 11*, 243–271.

Summary

This paper presents *FLMin*, a new method for extracting frequent patterns in social networks. Unlike traditional methods that focus solely on structural regularities, the originality of our approach is its ability to exploit both the structure and the node attributes to extract regularities, called "*frequent links*", in the links that connect nodes sharing common characteristics.