11e Journées Francophones d'Extraction et de Gestion des Connaissances

Brest

AVEC'2011

9e Atelier Visualisation et Extraction de Connaissances

le 25 Janvier 2011

François Poulet
Université de Rennes I - IRISA
Campus de Beaulieu
35042 Rennes cedex

Bénédicte Le Grand
Laboratoire d'Informatique de Paris 6
4, place Jussieu
75005 Paris

Préface

Les outils de visualisation contribuent à l'efficacité des processus mis en œuvre en extraction de connaissances en offrant aux utilisateurs des représentations intelligibles et facilitant l'interaction.

La visualisation intervient à différentes étapes de la chaîne de traitement : dans les phases amont pour appréhender les données et effectuer les premières sélections, lors du processus de fouille, et dans la phase aval pour évaluer les résultats obtenus et les communiquer. Du fait de l'importance croissante accordée au rôle de l'utilisateur en fouille de données, les outils de visualisation sont devenus des composantes majeures des logiciels qui s'utilisent de plus en plus en coopération étroite avec des méthodes automatiques.

La fouille visuelle de données ("Visual Data Mining") qui vise à développer des outils interactifs adaptés au traitement des données et des connaissances associées intègre par essence des concepts issus de disciplines diverses : perception visuelle, psychologie cognitive, métaphores de visualisation, visualisation scientifique ou d'information, etc.

Au fil des années, l'atelier a illustré le besoin et l'intérêt croissants pour la visualisation en extraction de connaissances, ainsi que le rôle essentiel de l'interactivité. Il a également permis d'aborder les problèmes délicats que constituent l'évaluation des méthodes de visualisation et le traitement des données volumineuses. Les neuf éditions de cet atelier ont permis de réunir des participants de profils extrêmement variés, abordant la fouille visuelle de données sous des angles différents et complémentaires. Cet atelier est chaque année l'occasion de rencontres et d'échanges enrichissants autour des dernières avancées liées à la visualisation et l'extraction de connaissances.

Cette neuvième édition s'inscrit dans la continuité, avec une attention toute particulière sur la visualisation et l'extraction de connaissances dans des grands réseaux d'interaction, tels que les réseaux sociaux. L'utilisation conjointe de visualisation et de techniques d'analyse est aussi illustrée cette année avec un exemple de *Visual Analytics*. L'utilisabilité des outils proposés reste également une priorité et plusieurs démonstrations de logiciels sont au programme de l'édition 2011 de l'atelier.

François Poulet
Bénédicte Le Grand

Les organisateurs tiennent à remercier les membres du comité de programme pour leur travail :

Michael Aupetit, CEA-LIST, Gif-sur-Yvette
Nadir Belkhiter, Université de Laval, Québec
Thanh-Nghi Do, Université de Can Tho, Vietnam
Jean-Loup Guillaume, LIP6, Université Pierre et Marie Curie - Paris 6
Philippe Lenca, Télécom Bretagne, Brest
Guy Melançon, LABRI, Bordeaux
Annie Morin, IRISA-TexMex, Rennes
Monique Noirhomme-Fraiture, FUNDP, Namur, Belgique
Nguyen-Khang Pham, Université de Can Tho, Vietnam
Michel Soto, LIP6, Paris

AVEC'2011

9e Atelier Visualisation et extraction de connaissances

Sommaire

Session 1

Session 2

EVARIST: un outil de monitoring du buzz et de l'e-reputation sur Twitter

Etienne Cuvelier*, Marie-Aude Aufaure*

* Équipe Business Intelligence
Laboratoire Mathématiques Appliquées aux Systèmes
École Centrale Paris,
Etienne.Cuvelier@ecp.fr, Marie-Aude.Aufaure@ecp.fr

Résumé. Dans le monde interconnecté actuel, la vitesse de diffusion des informations amène à une formation des opinions tendant vers de plus en plus d'immédiateté. En effet, les grands réseaux sociaux, en permettant le partage, et donc, la diffusion de l'information de manière quasi-instantanée, accélèrent aussi la formation des opinions concernant l'actualité. Cela fait de ces réseaux de formidables observatoires des opinions et aussi de l'e-réputation. Dans cet article nous proposons un prototype d'outil, basé sur les treillis de Galois, nommé EVARIST, qui permet à un utilisateur, institutionnel ou individuel, à partir d'un ensemble de mots clés de son choix (thème, marque, nom propre,...) de visualiser l'ensemble des termes les plus associés circulant à travers le réseau Twitter, et formant donc l'actualité brulante (buzz) concernant le sujet choisi.

1 Introduction

Depuis leur apparition, les blogs et les réseaux sociaux suscitent un intérêt croissant pour l'observation et la modélisation de l'opinion, comme le démontre, notamment, la session spéciale que leur consacrent les conférences TREC depuis l'édition de 2006 (Ounis et al. (2006)). La détection de sujets d'actualité brûlante fut ainsi une des tâches de l'édition 2009 de cette session blogs (voir Macdonald et al. (2009) pour une présentation). De même, les réseaux sociaux tels Facebook et Twitter, de par leurs fonctions de partage et de transfert de l'information devraient aussi permettre d'observer quasiment en temps réel la formation des opinions et, permettre ainsi de détecter les tendances. Par exemple Kramer (2010) utilise les mots exprimant les émotions dans les statuts Facebook des utilisateurs américains pour synthétiser un index modélisant le concept de "Bonheur National Brut".

Ces réseaux sociaux sont donc des lieux d'observation privilégiés pour de la formation des opinions, notamment à propos d'un sujet choisi, que cela soit une personne (personnal branding), une institution officielle ou un opérateur industriel. Dans le cas de l'e-reputation, l'observation du buzz et plus particulièrement du buzz négatif (bad buzz) concernant le sujet choisi est particulièrement importante. Nous avons testé cette observation sur le plus réactif de ces réseaux, à savoir, Twitter.

Cet article s'organise de la façon suivante : dans la section 2 nous détaillons succinctement comment fonctionne le réseau Twitter, quelles sont ses contraintes et ses conventions, et les

difficultés d'analyse qu'elles entrainent. Dans la section 3 nous introduisons l'analyse formelle de concepts et les treillis de Galois, qui permettent de contourner les difficultés évoquées. Enfin dans la section 4 nous exposons les principes base de notre outil EVARIST, ainsi que les résultats obtenus sur un petit ensemble d'information relatives au mot clé "e-réputation". Nous terminons avec les conclusions et perspectives de développement.

2 Twitter et Micro-Blogging

Twitter a été créé en 2006 dans le but de permettre à ses utilisateurs de partager facilement de courts messages textuels appelés *tweets* [1]. Le système ayant été initialement conçu pour partager les tweets via SMS, une limite de 140 caractères a été fixée à ces messages courts. Bien que le système soit aujourd'hui massivement utilisé via le web et via des applications développées pour ordinateurs ou smartphones, cette contrainte de 140 caractères n'a jamais été levée. L'aspect réseau social de cette plateforme réside dans son aspect orienté, et ses principes de base sont les suivants :

- un utilisateur peut, avec son compte Twitter, génèrer ou transmettre de l'information via un champ de saisie ;
- un utilisateur A peut suivre les tweets d'un utilisateur B en le signalant via l'interface ad hoc, et ce, sans que B ne doive suivre les tweets de A en retour.

Les utilisateurs qui suivent un compte Twitter A s'appellent ses *abonnés* ou *followers*, alors que les utilisateurs que A suit s'appellent ses *abonnements* ou *followings*. L'ensemble des tweets des abonnements d'un compte s'appelle sa *timeline*.

De par son principe de "micro-publications", Twitter permet un partage et une diffusion très rapides de l'information ainsi que des opinions concernant cette dernière. La croissance de ce service est actuellement importante et, au mois d'avril 2010, Twitter comptait presque 6 millions d'utilisateurs enregistrés, 300 000 nouveaux comptes par jour et, en moyenne, 55 millions tweets générés par jour (voir Bosker (2010)). Cette intense activité se traduit par une très grande réactivité par rapport aux faits d'actualité, et cette réactivité se révèle extrêmement intéressante pour l'analyse de la formation d'opinions. Ainsi O'Connor et al. (2010) ont montré qu'il existait une corrélation très importante entre trois indices existants, calculés via des enquêtes quotidiennes, concernant la confiance des consommateurs américains, les sondages Obama/Mc Cain pendant la campagne présidentielle américaine et ensuite l'appréciation du travail du président Obama, et les opinions formulées sur Twitter à propos de ces sujets. De manière plus prédictive Ritterman et al. (2009) ont montré que l'information circulant sur Twitter concernant la grippe aviaire permettait, associée à un modèle de prédiction de marché, de prédire plus efficacement l'opinion concernant la transformation de la grippe en pandémie.

Un certain nombre de conventions ont cours sur Twitter, nous allons en relever maintenant les principales, nécessaires à la compréhension minimale de la plateforme. La première de ces conventions est l'utilisation de l'arobase pour citer ou s'adresser à un utilisateur précis, ainsi dans l'exemple ci-dessous, l'utilisateur Jules s'adresse à l'utilisateur Jim en citant l'utilisatrice Catherine :

```
Jules: @Jim rendez-vous chez @Catherine à 22h00?
```

1. Gazouillis en anglais.

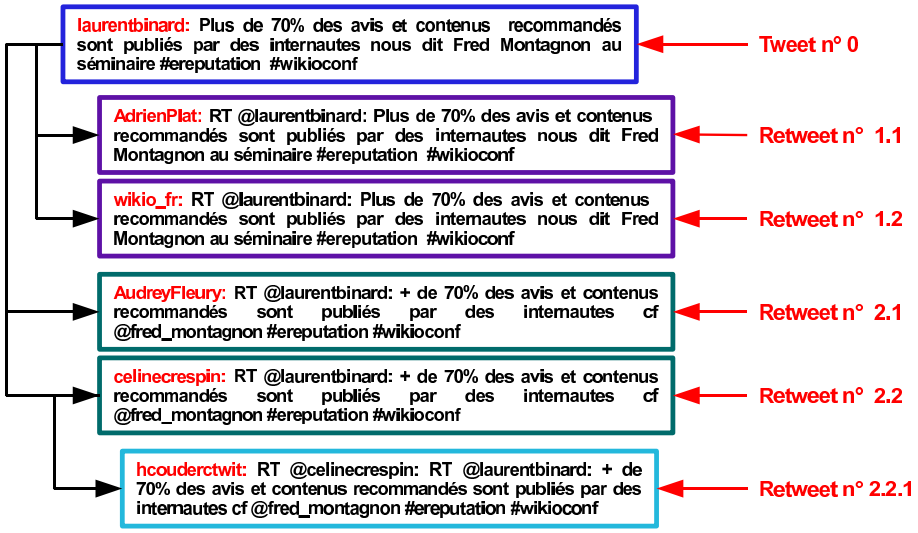

FIG. 1 – *Illustration du polymorphisme de l'information relayée.*

Les utilisateurs Jim et Catherine verront apparaitre ce tweet dans leurs timelines respectives. Une seconde convention est la pratique du *retweet ou RT*. Lorsqu'un utilisateur voit dans sa timeline une information qu'il veut partager avec ses abonnés, il utilisera la fonction retweet du service (web ou application) qu'il utilise, comme illustré dans l'exemple ci-dessous :

```
Jules: Inception est génial.
Jim: Bof RT @Jules: Inception est génial.
Catherine: LOL RT @Jim: Bof RT @Jules: Inception est génial.
```

Bien que la majorité des retweets soient annoncés par *"RT @"*, d'autres variantes et pratiques coexistent, comme cela a été très bien analysé dans Boyd et al. (2010). Par exemple certains utilisateurs éditent le retweet en ajoutant, par exemple un *(via @)* à la fin de celui-ci, comme dans l'exemple suivant :

```
Jules: http://www.google.com is awesome!
Catherine: RT @Jules: http://www.google.com is awesome!
Jim: http://www.google.com is awesome! (via @Jules)
```

Cette possibilité d'éditer un retweet, associée à la contrainte des 140 caractères crée ce que nous appellerons le *polymorphisme de l'information relayée* sur Twitter. Ce polymorphisme est illustré dans la figure 1. Dans cet exemple on voit que le tweet initial (tweet no 0) a été retweeté de plusieurs façons. On voit un premier groupe de retweets (no 1.1 et no. 1.2) dans lesquels le tweet initial est repris tel quel, et un second groupe de retweets (no 2.1 et no 2.2), où les utilisateurs ont modifié légèrement le retweet. De plus un des retweets de ce deuxième groupe est lui même retweeté tel quel (tweet no 2.2.1). Ce polymorphisme pose un réel problème lorsque l'on veut mesurer la popularité d'une information, car si l'on se limite à compter le nombre de retweets non-modifiés, on omettra de comptabiliser une partie des retweets, qui malgré leurs modifications véhiculent essentiellement la même information.

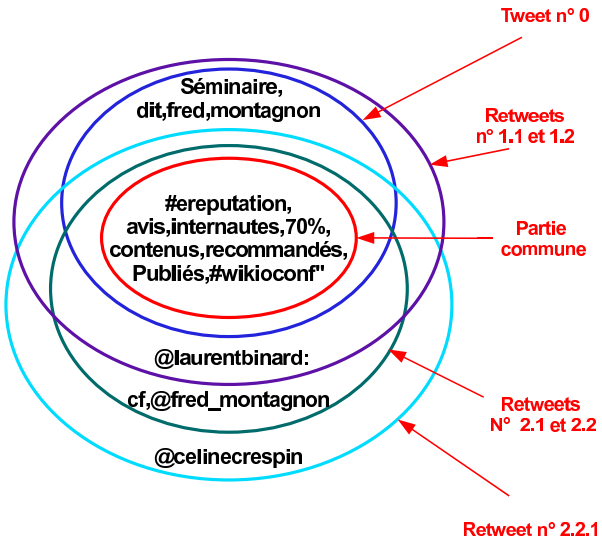

FIG. 2 – *Vue ensembliste du polymorphisme de l'information relayée.*

Néanmoins, si l'on fait abstraction des mots de liaisons et des signes de ponctuations (en l'occurrence dans notre exemple : plus, +, de, des, et, sont, par, nous), que l'on se restreint aux mots signifiants, alors on peut représenter l'information relayée sous forme ensembliste, comme l'illustre la figure 2. On voit dans cette représentation que les inclusions successives permettent de visualiser la partie commune à tous les tweets, le noyau de l'information relayée. L'inclusion définit aussi une relation d'ordre partiel qui peut être représentée par un diagramme de Hasse comme dans la figure 3. L'établissement d'un tel diagramme permet d'établir, à la fois, quels sont les mots communs à un ensemble de tweets, ainsi que les différentes formes sous lesquelles une même information a été relayée. L'analyse formelle de concepts et les treillis de Galois permettent exactement de structurer l'information sous cette forme.

3 Analyse Formelle de Concepts - Treillis de Galois

L'*analyse formelle de concept*, en anglais *Formal Concept Analysis, FCA*, (Wille (1980)) est basée sur les *Treillis de Galois* (Barbut et Monjardet (1970) et Birkhoff (1940)), qui peuvent être utilisés pour la classification conceptuelle (Carpineto et Romano (1993) et Wille (1984)).

Un treillis de Galois permet de regrouper, de façon exhaustive, des objets en classes, appelées *concepts*, en utilisant leur propriétés partagées. Un treillis est classiquement basé sur une matrice booléenne, appelée *matrice de contexte* et notée C, dont les lignes représentent un ensemble d'*objets* O que l'on souhaite décrire, et les colonnes, un ensemble d'*attributs* A que ces objets ont ou non et, qui permettent donc la description des dits objets. Nous utiliserons, pour introduire ces treillis, un exemple simple que l'on peut trouver dans Wolff (1993). Supposons que nous avons une description des espèces animales suivantes (voir table 1) : Lion, Moineau, Aigle, Lièvre, Autruche, Abeille et Chauve-souris. Description basée sur la liste des propriétés que les animaux de l'espèce possèdent ou non : Sont-ils prédateurs ? Volent-ils ? Sont-ce des

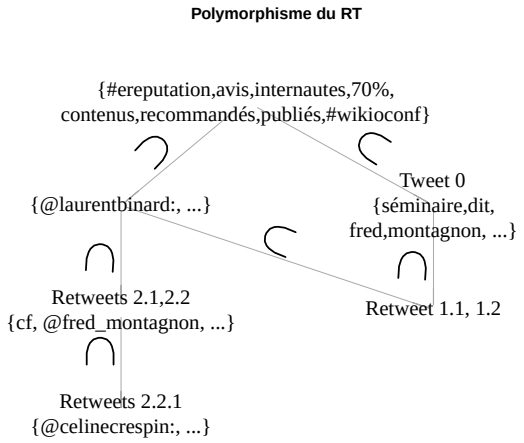

FIG. 3 – *Analyse du polymorphisme du retweet de l'information.*

	Prédation	Vol	Oiseau	Mamifère
Lion	x			x
Moineau		x	x	
Aigle	x	x	x	
Lièvre				x
Autruche			x	
Abeille		x		
Chauve-souris		x		x

TAB. 1 – *Table de contexte pour les espèces animales.*

oiseaux ? Sont-ce des mammifères ? La possession de la propriété $a \in A$ par l'objet $o \in O$ traduit l'existence d'une relation I entre eux : aIo. L'existence de cette relation I entre O et A est matérialisée dans la matrice de contexte C par, soit la valeur "vrai" (et "faux" sinon) soit par une marque quelconque (et rien sinon) . Le triplet $K = (O, A, I)$ est appelé un *contexte formel* ou simplement un contexte.

L'*intention* d'un ensemble $X \subset O$ est l'ensemble des attributs possédés conjointement par tous les objet de X et, est donnée par la fonction f :

$$f(X) = \{a \in A | \forall o \in X, oIa\}. \quad (1)$$

Inversement l'*extension* d'un ensemble $Y \subset A$ est l'ensemble des objets à posséder conjointement touts les attributs de Y et, est donnée par la fonction g :

$$g(Y) = \{o \in O | \forall a \in Y, oIa\}. \quad (2)$$

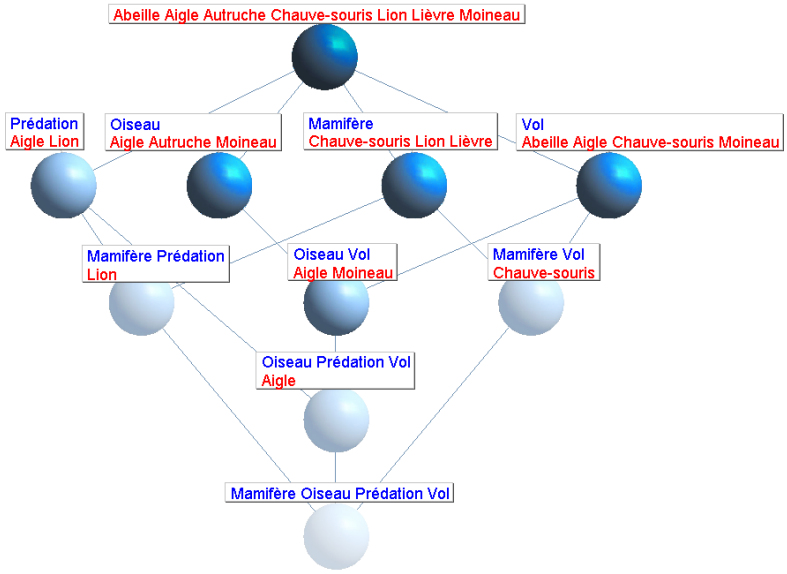

FIG. 4 – *Affichage plein du treillis de Galois des espèces animales de la table 1.*

Le couple (f, g) est appelé une *connexion de Galois*.

Un *concept* est tout couple $C = (X, Y) \subset O \times A$, tel que *les objets de X soient les seuls à posséder tous les attributs de Y*, en d'autres termes $X \times Y$ forme, à deux permutations près de O et de A, un rectangle maximal dans C, c'est-à-dire que

$$f(X) = Y \ \& \ g(Y) = X. \tag{3}$$

Pour illustrer cette notion de concept on peut observer dans la table 1 que l'ensemble $X = \{$*Moineau, Aigle*$\}$ donne un concept car $f(X) = \{$*Vol, Oiseau*$\} = Y$ et $g(Y) = X$, et ce concept est donc $(\{$*Moineau, Aigle*$\}, \{$*Vol, Oiseau*$\})$, alors que l'ensemble $X' = \{$*Lion, Lièvre*$\}$ ne donne pas un concept car $f(X') = \{$*Mammifère*$\} = Y'$ and $g(Y') = \{$*Lion, Lièvre, Chauve-souris*$\}$, mais par contre ce dernier ensemble, lui, donne le concept $\{$*Lion, Lièvre, Chauve-souris*$\}, \{$*Mammifère*$\}$.

Le *treillis de Galois* est le *poset* de concepts L muni de l'ordre partiel suivant $\leq$:

$$(X_1, Y_1) \leq (X_2, Y_2) \Leftrightarrow X_1 \subseteq X_2 (\text{ or } Y_1 \supseteq Y_2). \tag{4}$$

Le treillis de Galois est noté $T = (L, \leq)$ et, est représenté à l'aide d'un *diagramme de Hasse* comme dans les figures 4 et 5 pour les espèces. Deux types d'affichage existent pour les labels des concepts, l'affichage plein et l'affichage réduit. Dans l'affichage plein, tous les objets et attributs d'un concept sont affichés, comme dans la figure 4. Dans l'affichage réduit, les attributs et les objets ne sont affichés qu'une seule fois, la première fois qu'ils sont rencontrés en parcourant le treillis, en partant du haut pour les attributs, et en partant du bas pour les objets, comme c'est le cas dans la figure 5.

Le calcul du treillis peut être effectué à l'aide, par exemple, de l'algorithme de Bordat (1986), qui calcule récursivement tous les concepts à partir du concept $(\emptyset, f(\emptyset))$, en calculant

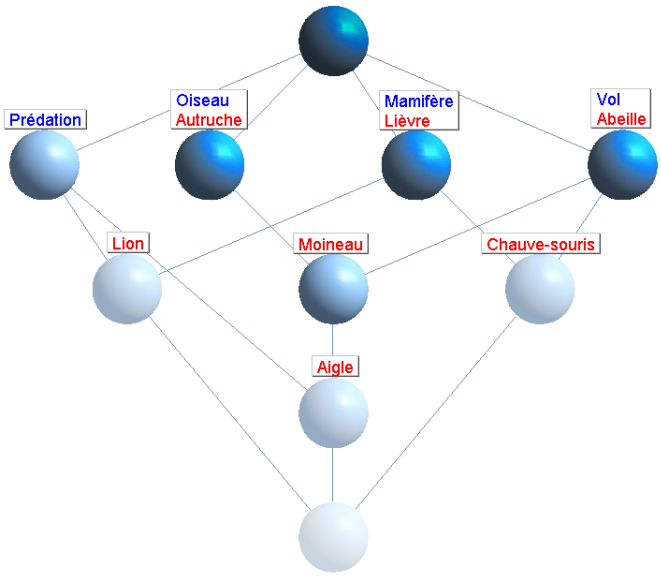

FIG. 5 – *Affichage réduit du treillis de Galois des espèces animales de la table 1.*

pour chaque concept trouvé l'ensemble de ses concepts-fils. Pour voir une revue des autres algorithmes pour la génération de treillis, voir l'article de Kuznetsov et Obedkov (2001) qui effectue aussi une comparaison des performances.

Un des avantages non négligeable de la classification basée sur les treillis est que pour une table de contexte donnée le treillis résultant est unique (pas d'instabilité à l'exécution), et il est exhaustif (tous les concepts existants s'y trouvent). Dans le cas qui nous occupe cette classification va nous permettre de retrouver tous les groupes de mots présents dans un groupe de tweets et de les représenter de façon similaire à la figure 3.

4 E-buzz Monitoring

Dans ce qui suit nous proposons d'analyser un groupe de tweets, afin d'y repérer les termes et groupes de termes les plus tweetés dans le groupe considéré. Pour ce faire nous proposons une démarche en quatre étapes :

1. Récupération des tweets contenant un ou plusieurs mots clés ;
2. Nettoyage des tweets (suppression des mots de liaisons, ponctuations,...) ;
3. Établissement d'une table de contexte où les objets sont les tweets et, les attributs les mots contenus dans ces derniers ;
4. Calcul du treillis de Galois correspondant ;
5. Visualisation des résultats.

Pour illustrer notre technique nous utiliserons un ensemble de 50 tweets récupérés en faisant une recherche sur le mot clé "#ereputation". Nous donnons ci dessous, en guise d'illustration les 5 premiers tweets de cet ensemble :

```
Tweet 1: overclub: #ereputation : votre avis sur les multiples
solutions de veille... d'après vous quel est l'outil le plus efficace ?
Tweet 2: AudreyFleury: #eReputation Les internautes veillent sur leur
eréputation Stratégies http://ow.ly/1a7fWM
Tweet 3: AudreyFleury: RT @laurentbinard: + de 70% des avis et contenus
recommandés sont publiés par des internautes cf @fred_montagnon
#ereputation #wikioconf
Tweet 4: hcouderctwit: RT @celinecrespin: RT @laurentbinard: + de
70% des avis et contenus recommandés sont publiés par des internautes
cf @fred_montagnon #ereputation #wikioconf
Tweet 5: wikio_fr: RT @laurentbinard: Au séminaire  #wikioconf, Serge
Alleyne, fondateur de  #nomao, annonce et présente sa solution  de
#ereputation locale avec  #wikiobuzz...
```

L'application des étapes 1 à 4 ne pose pas de problème sur un treillis de taille modeste, alors que la cinquième, la visualisation des résultats, elle, est moins évidente. Dans la figure 6 nous affichons l'ensemble du treillis en donnant à chaque concept une surface qui soit proportionnelle au nombre de tweets qu'il contient. On constate que malgré sa petite taille (59 concepts trouvés) il est difficile d'afficher tout les concepts proportionnellement à leur taille et, simultanément, l'ensemble de leurs attributs de façon lisible, et ce même en utilisant l'affichage réduit des attributs. Pour réduire le nombre de concepts à afficher, nous pouvons ne sélectionner que les concepts dont la taille relative (nombre d'objets du concept divisé par le nombre d'objets dans la table de contexte) dépasse un seuil choisi, ce qui est cohérent avec le concept de buzz, car ce sont les groupes de mots les plus tweetés. C'est ce qui est fait dans la figure 7 avec un seuil de 10%, mais cela n'augmente pas suffisamment la lisibilité, car les attributs de notre treillis sont formés de mots ou groupes de mots plus ou moins longs. Un autre type de visualisation s'avère donc nécessaire pour ce groupe de concepts avec les mots les plus retweetés. On peut bien entendu essayer d'afficher les mots contenus dans les différents concepts à l'aide de nuage de tags, en donnant aux mots du concept une taille proportionnelle à l'importance de celui-ci. Néanmoins, même s'il existe des solutions pour afficher les tags associés l'un près de l'autre comme dans Kaser et Lemire (2007), on perd avec cet affichage les liaisons d'inclusion entre un concept et ses sous-concepts. De plus un sous-concept pouvant dépendre de plusieurs super-concepts, cela risque de complexifier la tâche de regroupement des tags proches. Nous proposons donc d'afficher les concepts les plus importants sous forme de nuage de tags, proportionnels et en réseau, c'est-à-dire que les liaisons entre les différents concepts seront matérialisées par des arrêtes. Ces arrêtes seront dirigées, allant du concept vers ses sous-concepts. Nous effectuons le placement des noeuds à l'aide de la méthode Fruchterman et Reingold (1991), car cette méthode optimise l'écartement des noeuds et permet ansi d'augmenter la lisibilité des tags. Enfin, pour renforcer une lecture allant du plus général au plus particulier, nous avons décidé d'ajouter une allégorie topographique, similairement aux cartes topigraphiques proposées dans Fujimura et al. (2008). Pour ce faire, à chaque point du graphique résultant, nous ajoutons un niveau, les différents niveaux étant représentés à l'aide des classiques courbes de niveaux. Pour la mise aux points des niveaux, nous utilisons un mélange de gaussiennes à deux dimensions transformées, auxquelles nous donnons comme centres les coordonnées des centres des tags, et comme écart-types les largeurs et hauteurs des dits tags. Enfin, pour donner des hauteurs qui soient proportionnelles aux tailles des concepts exactement aux centres des tags, nous normalisons les hauteurs des gaussiennes en les multipliant par les

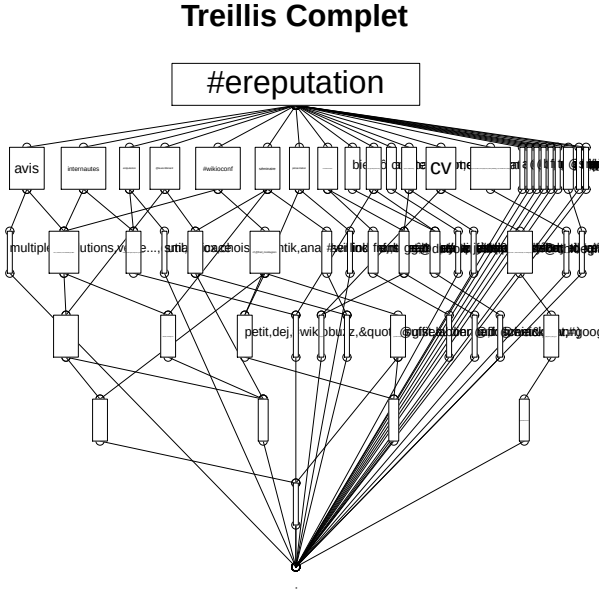

FIG. 6 – *Le treillis de Galois des tweets.*

écart-types et ensuite par hauteurs désirées. Le mélange résultant est la fonction topigraphique T :

$$T(x,y) = \sum_{i=1}^{k} \frac{s_i}{2\,\pi}\,\mathrm{e}^{-\dfrac{(x-x_i)^2\,(x-y_i)^2}{2 l_i^2 h_i^2}} \tag{5}$$

où :

- k représente le nombre de concepts affichés,
- x_i et y_i représentent les coordonnées du i^e concept,
- l_i et h_i représentent la largeur et la hauteur du tag du i^e concept,
- s_i représente la taille du i^e concept.

Bien entendu, comme nous avons changé les volumes sous les surfaces, notre fonction topigraphique T n'est plus une densité de probabilité, mais cette propriété ne nous est pas nécessaire ici.

Le résultat final peut être vu en figure 8. Dans cette figure, à partir du concept représentant le mot clé de départ *{#ereputation}*, on voit que les sous-concepts les plus importants sont *{avis}*, *{internautes}* et *{#wikiconf}*, et que ces trois concepts contiennent aussi le concept *{70%, contenus, recommandés, publiés}*, alors que seul le concept *{#wikiconf}* contient le concept *{cf, @fred_montagnon}*. D'autre part on voit trois concepts affichés indépendamment des premiers : *{c'est}*, *{@laurentbinard :}* et *{daily, read, twitter, newspaper, http ://bit.ly/d, (19, contributions, todays)}*. Et enfin le concept *{cv}* qui dépend du concept *{c'est}*. L'idée de cette visualisation est de laisser glisser le regard du lecteur des "sommets" (les concepts les

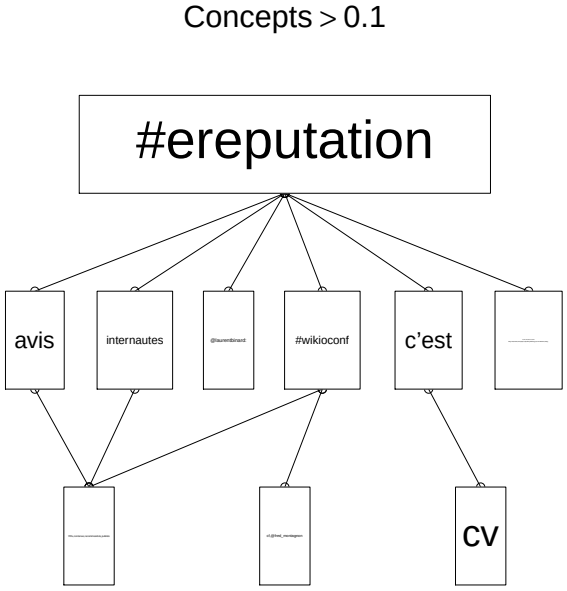

FIG. 7 – *Les concepts comprenant plus de 10% des tweets.*

plus généraux) vers les "vallées" (les concepts les plus particuliers). La constructions et les affichages des treillis et des cartes de tags en réseau topigraphique ont été effectuées dans l'environnement statistique R Development Core Team (2010), et pour les parties treillis, à l'aide du package *galois* développé par nos soins.

5 Conclusions et perspectives

Dans cet article nous avons présenté une nouvelle technique de monitoring du buzz et de la e-réputation sur la plateforme de micro-blogging, Twitter. Cette technique se base sur les treillis de Galois, et propose comme visualisation des concepts résultants, un nuage de tags proportionnels en réseau topigraphique. Cet affichage, limité aux concepts les plus importants, permet d'afficher les tags constituant les concepts de manière plus lisible que si l'on affiche directement le treillis. Son idée est de faire "glisser" le regard du lecteur, des concepts les plus généraux, affichés aux "sommets" vers les concepts plus particuliers placés dans les "vallées", les flèches du réseaux servant de "pistes" pour guider vers les concepts liés.

Bien qu'EVARIST soit encore à l'état de prototype, un certain nombre d'améliorations sont envisageables et envisagées. La première c'est l'ajout de l'interactivité afin de permettre à l'utilisateur de sélectionner le sous-concept qu'il désire développer. On envisage ensuite de "développer" les URL réduites (bit.ly, is.gd, tinyURL,..) afin de ne pas comptabiliser sous deux concepts différents une même URL qui aurait été réduite via deux services différents. Lors de l'étape qui consiste en la suppression des mots de liaisons et des signes de ponctuations afin

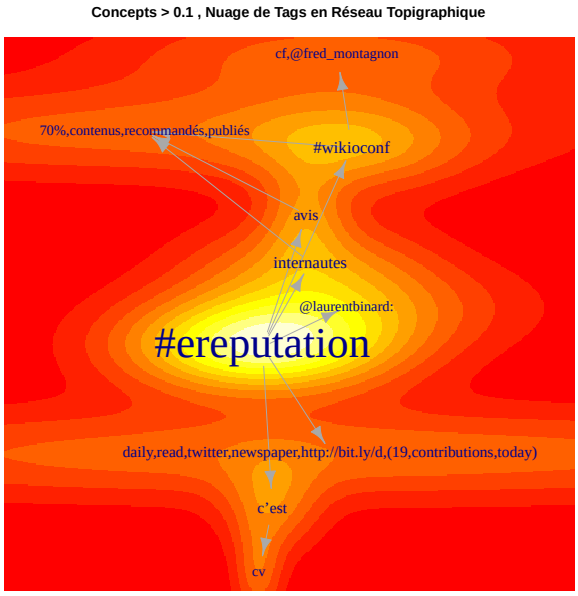

FIG. 8 – *Le réseau topigraphique des concepts comprenant plus de 10% des tweets.*

d'établir la table de contexte, un traitement particulier devrait être réservé aux smileys, qui sont en fait signifiants. Dans une prochaine étape, nous prévoyons aussi d'utiliser l'analyse de sentiments pour évaluer la positivité ou négativité des concepts, notions importantes dans le cadre de la surveillance de la réputation électronique. Enfin le support de plusieurs langues se révèle un challenge important mais intéressant pour le développement d'un tel outil.

Références

Barbut, M. et B. Monjardet (1970). *Ordre et classification, Algebre et combinatoire, Tome 2*. Hachette.

Birkhoff, G. (1940). *Lattice Theory*, Volume 25. New York : American Mathematical Society.

Bordat, J. (1986). Calcul pratique du treillis de galois dŠune correspondance. *Mathématique, Informatique et Sciences Humaines 24*(94), 31–47.

Bosker, B. (2010). Twitter user statistics revealed. http ://www.huffingtonpost.com/.

Boyd, D., S. Golder, et G. Lotan (2010). Tweet, tweet, retweet : Conversational aspects of retweeting on twitter. In *Proceedings of the 43rd Hawaii International Conference on Social Systems (HICSS)*.

Carpineto, C. et G. Romano (1993). Galois : An order-theoretic approach to conceptual clustering. In *Proc. Of the 10th Conference on Machine Learning, Amherst, MA, Kaufmann*, pp. pp. 33–40.

Fruchterman, T. et E. Reingold (1991). Graph drawing by force-directed placement. *Software - Practice and Experience 21*(11), 1129–1164.

Fujimura, K., S. Fujimura, T. Matsubayashi, T. Yamada, et H. Okuda (2008). Topigraphy : visualization for large-scale tag clouds. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, New York, NY, USA, pp. 1087–1088. ACM.

Kaser, O. et D. Lemire (2007). Tag-cloud drawing : Algorithms for cloud visualization. In *WWW2007 Workshop on Tagging and Metadata for Social Information Organization,*, Banff, Alberta.

Kramer, A. D. I. (2010). An unobtrusive behavioral model of Ş gross national happinessŞ. In *Proceedings of the 2010 conference on Human Factors and Computing Systems (CHI 2010)*.

Kuznetsov, S. O. et S. A. Obedkov (2001). Comparing performance of algorithms for generating concept lattices. In *Concept Lattices-based Theory, Methods and Tools for Knowledge Discovery in Databases (CLKDD'01)*, Stanford, July 30, 2001.

Macdonald, C., I. Ounis, et I. Soboroff (2009). Overview of the trec 2009 blog track. In *NIST Special Publication 500-278 : The Eighteenth Text REtrieval Conference Proceedings (TREC 2009)*.

O'Connor, B., R. Balasubramanyan, B. R. Routledge, et N. A. Smith (2010). From tweets to polls : Linking text sentiment to public opinion time series. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media, Washington, DC*.

Ounis, I., C. M. M. de Rijke, G. Mishne, et I. Soboroff (2006). Overview of the trec 2006 blog track. In *NIST Special Publication 500-272 : The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)*, Volume 272, pp. 17–31.

R Development Core Team (2010). *R : A Language and Environment for Statistical Computing*. Vienna, Austria : R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Ritterman, J., M. Osborne, et E. Klein (2009). Using prediction markets and twitter to predict a swine flu pandemic. In *1st International Workshop on Mining Social Media*.

Wille, R. (1980). *Restructuring lattice theory, Ordered sets I*. Rival.

Wille, R. (1984). Line diagrams of hierarchical concept systems. *Int. Classif. 11*, 77–86.

Wolff, K. E. (1993). A first course in formal concept analysis - how to understand line diagrams. In F. Faulbaum (Ed.), *SoftStatt'93, Advances in Statistical Software 4, 429-438.*

Summary

In the actual interconnected world, the speed of broadcasting of information leads the formation of opinions towards more and more immediacy. Big social networks, by allowing distribution, and therefore broadcasting of information in a almost instantaneous way, also speed up the formation of opinions concerning actuality. Then, these networks are great observatories of opinions and e-reputation. In this article we propose a prototype of tool, based on Galois lattices, which allows an user, institutional or individual, from a chosen set of key words (topic, mark, proper name) to show all the most linked terms circulating across Twitter, and forming therefore the hot topics (Buzz) concerning the chosen subject.

Correspondence Analysis for Exploration of Telecommunication Data

Nguyen-Khang Pham*, Hung-Thang Pham**

*Can Tho University
1, Ly Tu Trong Street, Can Tho, Viet Nam
pnkhang@cit.ctu.edu.vn
http://www.ctu.edu.vn
**Can Tho – Hau Giang Telecommunications
11A, Phan Dinh Phung Street, Can Tho, Viet Nam
thangph.cthg@vnpt.vn
http://www.vnpt.vn

Summary. We propose, in this paper, an interactive graphical tool, which allows us to analyze, visualize and extract knowledge from telecommunication data using Correspondence Analysis. Telecom companies have a very large volume and valuable data source on consumer information and call record details (CRD's). The CRD's not only indicate when a service is used, but also said that it is used. How we can point out what customers need, and why they should use a certain service. CRD's also said customers who are unhappy, and customers who bring more profit than others. Therefore, challenges that telecom companies do face is to understand the customer, to study their habits of using telecommunication services. Correspondence Analysis (CA) is originally for analysis of contingency tables. In order to adapt CA on telecommunication data, they are coded in a form of contingency table crossing customers and telecommunication services. Applying successively CA on this table allow us to explore groups of customers who have the same habit of using telecommunications services. First, the results of the first CA are visualized. Then the user selects a group of customers and performs the second CA on the new contingency table. This procedure can continue until a "pure" group of customers is discovered. An application to the CRD's database of Can Tho – Hau Giang Telecommunications shows the interest of our tool for extracting knowledge from telecommunication data.

1 Introduction

Nowadays, information has become a powerful resource of modern business. More than ever, the business organization in the world have focused on developing information systems as a key source of creating competitive advantage and improve responsiveness to market. Not outside the telecommunications industry trend, telecom companies have very valuable data source that is a large volume of data on consumer information and calls record detail (CRD's), every day millions of calls recorded in exchanges with the first goal is to charge the customer and network management. The call records not only indicate when a service is used, but also said that service is used, how it can point out what clients need, and why cus-

tomers should buy service, CRD's also said customers who are unhappy, customers who bring more profit. Therefore challenges that telecom companies do face is to understand their customers. Learn to know the customer habits of using telecommunication services. Some policies for customer care will be taken decision.

Monitoring the use of telephone customers, we will find that the customers use the various services, these services are called again for many months to create the habit of using, the subgroup of customers have the same habit of using the service. Instead of examining every customer, we only focus on one group of customers with the best response rate, so we can save advertising costs and increase promotional efficiency. The question that needs to be a tool to analyze the practice of using telecommunication customers, from which the policy-oriented promotions, advertising, customer care and development of new services by demand for use by customers.

This article mentions a new approach in the field of Telecommunication, using a statistical method - the correspondence analysis (CA) - to analyze the practice of using telecommunication customers, basic difference of this method is observed on the graph to find the hidden rules.

2 Method

CA is a classical exploratory method for the analysis of contingency tables. It was proposed first by J. P. Benzécri (1973) in a linguistic context, *i.e.* textual data analysis. Theoretical developments and various applications can be found in the books of M. Greenacre (1984, 2007). CA on a table crossing words and documents (texts or images) allows answering the following questions: is there any proximity between certain words? Is there any proximity between certain documents? Is there any link between certain words and certain documents? CA, like most factorial methods, uses singular value decomposition (SVD) or eigen decomposition of a particular matrix and translate two-way tables into more readable graphical forms in low-dimensional space.

One fundamental concept of CA is that of a profile which is a set of frequencies divided by their total. In analyzing a frequency table, we can look at the relative frequencies for rows or for columns, called row or column profiles respectively. Each row (resp. column) has its own mass. And these masses are proportional to the marginal sums of the table. If the table has r rows and c columns, each row (resp. column) profile can be represented in a (c - 1) (resp. (r - 1)) dimensional space. The inertia of the row-profiles cloud and the inertia of the column profiles cloud are both equal to the chi-square statistics of the two-table divided by the total number of observations.

In most applications, the table of interest has many rows and columns. So r and c are large and it becomes necessary to reduce the dimensionality of the points. The objective of CA is to discover the subspaces where the cloud of projected row (resp. column) profiles have the largest inertia. This is done through a SVD or an eigen decomposition. The accuracy of display is measured by the percentage of inertia of the projected profiles related to the original cloud inertia. In addition, CA provides relevant indicators for the interpretation of the axes which define a subspace such as the contribution of a word or a document to the inertia of the axis or the representation quality of a word and/or document on an axis (Morin, 2004).

CA has successfully been used too for content based image analysis (Pham et al, 2009a) or image retrieval (Pham et al, 2009b).

3 Interactive exploration of telecommunication data

3.1 Preprocessing

Due to the nature of telecommunication data (some customers never make a call, other use a single service) we have proposed some criteria for constructing the contingency table.

- Data are collected every three months. When customers will need to use the service, it needs to be repeated in months to form the habit of using the service and also reflects the customer relation. A period of 3 months is sufficient to form the habit of using customer service.
- Customers who use less than 3 services will be ignored. It is easy to analyze these customers by focusing on the services used.
- Only services which are used at least 3 times are selected.

We have, after this step, a contingency table crossing customers and services. The element f_{ij} of this table represent how often the customer *i* uses the service *j*.

3.2 Projection on factorial plan

In CA, customers and services are displayed on the same plane. Here, a dot "customer" is displayed as a *red circle*; and a dot "service" as a *blue circle*. User can select one or a group of customers (or services) by pointing on it.

To help the user for interpretation, the label of the service is displayed besides the corresponding dot "service". This gives us immediately a general summary of the group of services, *e.g.* international call.

For focusing on the interesting customers and/or services, we display only the customers and/or services whose contributions to inertia are high, usually 2 or 3 times the average contribution. Other customers/services will not be displayed. The total inertia on one axis is equal to the eigenvalue associated to this axis. The threshold is easy to determine. User can also change displayed axes (hence plan) in order to discover more group of customers/services. The choice of displayed axes could be done using the *quality of representation* of customer/service variables. The quality of representation of a projected point *i* (service *i*) on the axis *j* is the square cosine of the angle between the original point *i* and its projection on axis *j*. If the square cosine is close to 1, this means that the position of the projected point is close to the original position.

3.3 Hierarchical CA by visualization

We introduce here a method which allows us to discover hierarchical groups of services. The method is based on an interesting property of CA: the simultaneous representation of lines (customers) and columns (services). When CA projects customers and services on the same plan, their association is interpreted by the distance (in the plan) among them. The points "services" around a point or a group of point "customers" indicate that there is a strong relation between these customers and services. We can apply another CA on these

points only. This results some interesting subgroups of services. This procedure can be repeated on the subgroups in order to discover "*sub subgroups*" and so on.

Focusing on a group of customers/services allows us to discover other factorial axes (hence other groups of customers/services) which could be hidden before because of the influence of axes found by the first CA.

4 Experimental results

4.1 Datasets

We have experimented on two datasets extracted from CRD's of Can Tho - Hau Giang Telecommunications. The both dataset are preprocessed by the procedure showed in Fig. 1.

```
Get all services used at least by two customers
FOR each customer who uses at least two services
    Compute the total revenues in the six months
    IF the total revenues is greater than 1,000,000 VND THEN
        Keep this customer
    ELSE
        Ignore this customer
```

FIG. 1 – *Procedure for selecting customers and services*

The first dataset consists of CRD's from January to June 2008 and the second from July to December 2008.

4.2 "January to June 2008" Dataset

4.2.1 Grouping customers

After the preprocessing step, we kept 9,602 customers and 56 services. Fig 2 displays customers and services on axes 2 and 4. In this figure, it is easy to find 4 groups of customers and their associated services. Group 1 consists of the customers who use the service 177 for inter-province calls. Similarly, group 2 corresponds to the customers using the service 171, Group 3 and 4 correspond to the customers who use the "intra province call" service and "local call" service respectively.

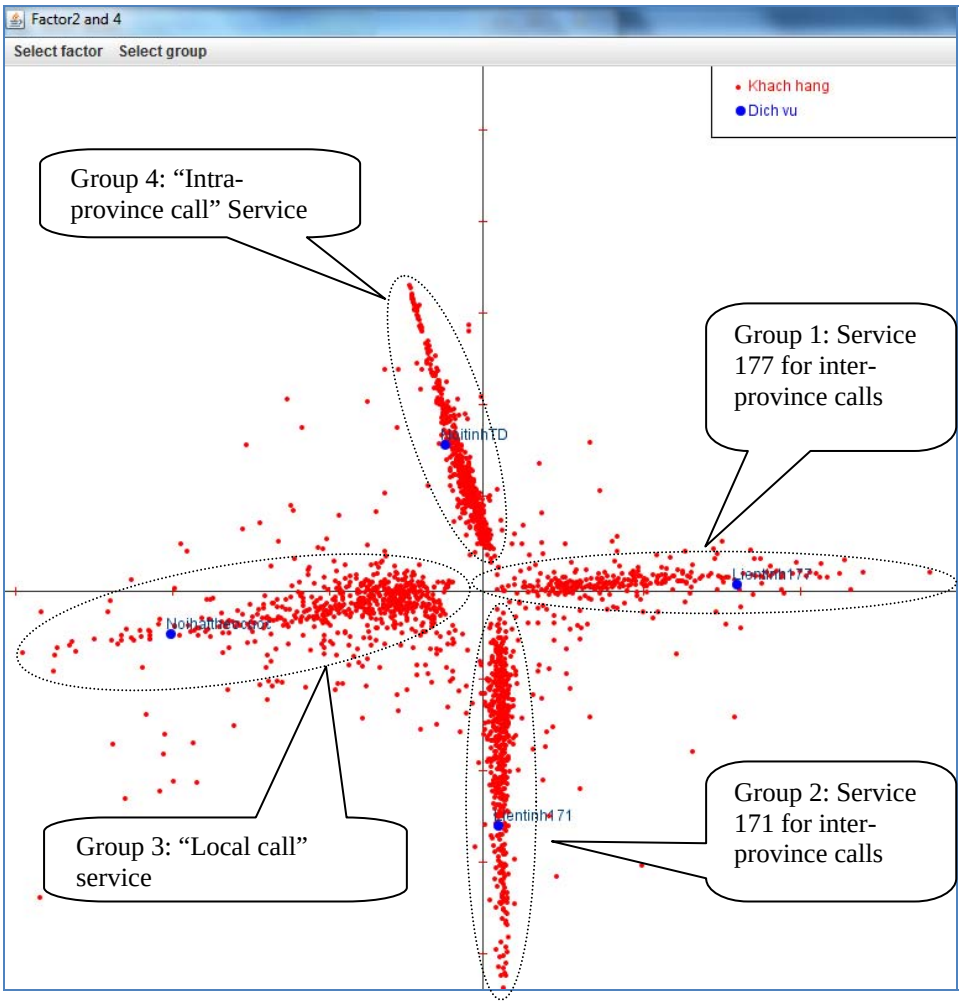

FIG. 2 – *Visualization of customers and service on the axes 2 and 4*

With this result, Can Tho – Hau Giang Telecommunication can propose some policies for better caring their customers

- For customers using the service 177 of SPT (another telecom company), reduce service charges 171 (the service of Can Tho – Hau Giang Telecom, Viet Nam Post and Telecommunications - VNPT) to drive customers to use the services 171.
- For customers using 171: these customers are using the service of VNPT, to maintain usage habits of customers, we can apply the discount policy for customers per month.
- For the customers using “inter province call” service: reduce charges for 171 or reduce charges for calls to the mobile numbers of Vinaphone (the service of Can Tho – Hau Giang Telecom).

Grouping customers by their behavior of using services is important for the company to divide customers into groups for better care. The analysis of the first six months of data will help the adjustment of customer care policies in the last six months.

To facilitate the observations, we have exported the data into four groups of files and plot the use of services in each group.

4.2.2 Coloring data

To better interpret the association of customers and other information as management department, type of customer: particular, business, post office,… we can colorize customer by associated information.

Fig. 3 shows points "customers" colorized by management department.

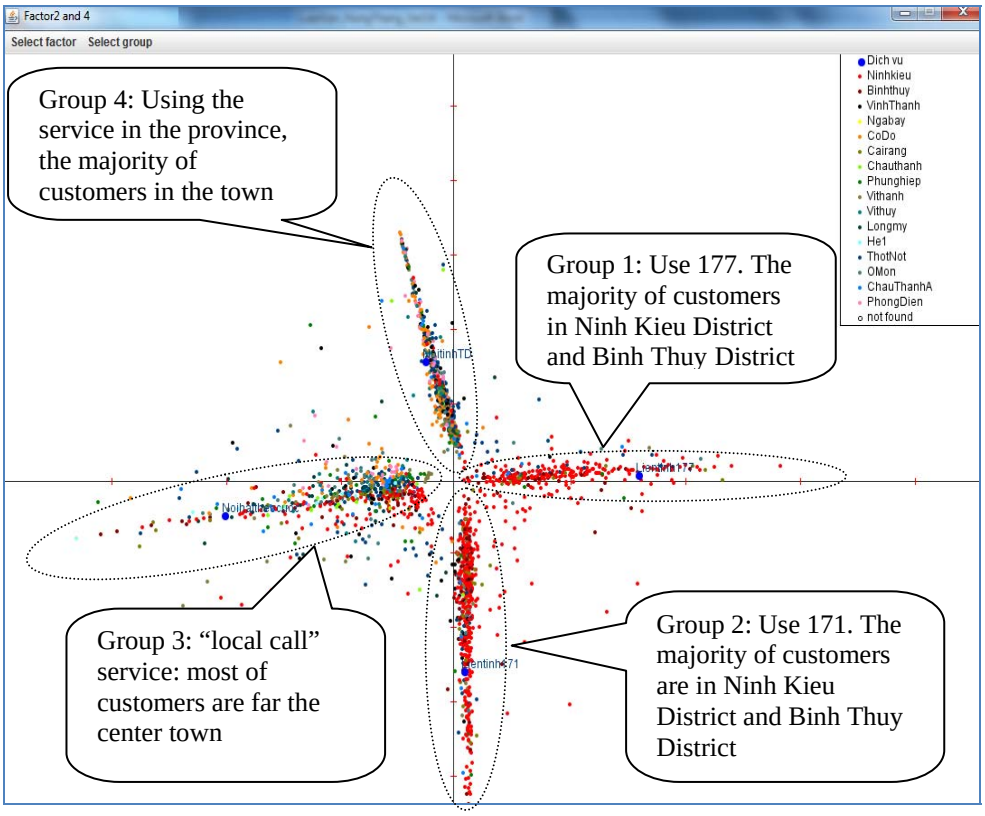

FIG. 3 – *Customers colorized by management department*

In this figure, we found that:

- The group of customers using the service 171 (group 2) and 177 (group 1) are majority at Ninh Kieu District and Binh Thuy (main districts of Can Tho). Two groups of customers in these districts are known the most use of “inter province call” service.
- Group 3 used by the local service, mostly in districts such as Chau Thanh, Chau Thanh A, Phung Hiep (far Can Tho city)
- Group 4 used in the province, the majority of customers are in districts such as Phung Hiep, Thot Not (near Can Tho city)

In short, when we see color matched two groups: regular intercity calls 171 and 177, the majority of customers in this group at Ninh Kieu District and Binh Thuy District. Local groups often referred to customers in the province. The majority of these customers in this group are in the districts such as: Phung Hiep, Chau Thanh, Thot Not (near Can Tho city).

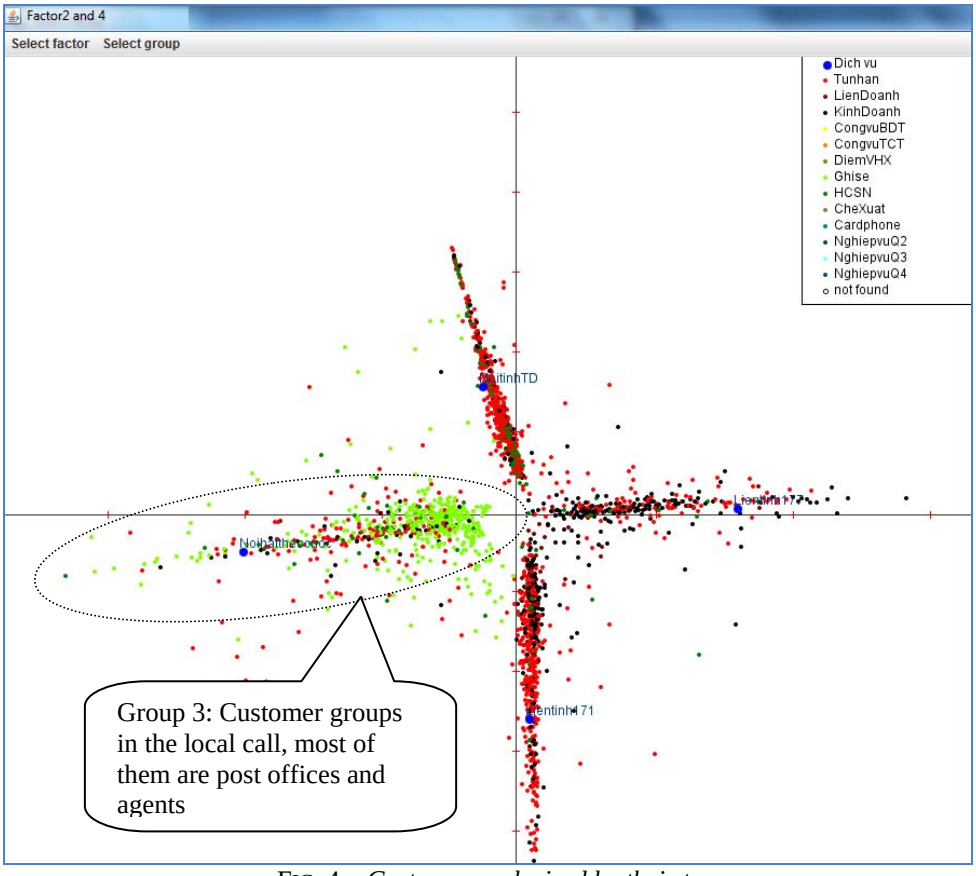

FIG. 4 – *Customers colorized by their type*

Fig. 4 display points “customers” colorized by their type. In the group of customers using local services (group 3), we found that most of them are post offices or record counters. In

group 1 and group 2, customers using 171 and 177 are mostly business and joint venture. Group 4 corresponds to customers using the service in the province; most of them are the particular customers.

Other groups of customers/services can be found by changing displayed axes or reanalyzing a group of customers and services. In this way, a hierarchy of groups of customers/services can be discovered. For instance, in Fig. 5 we focus only on customers near the origin, and perform another CA. Results are shown in Fig. 6.

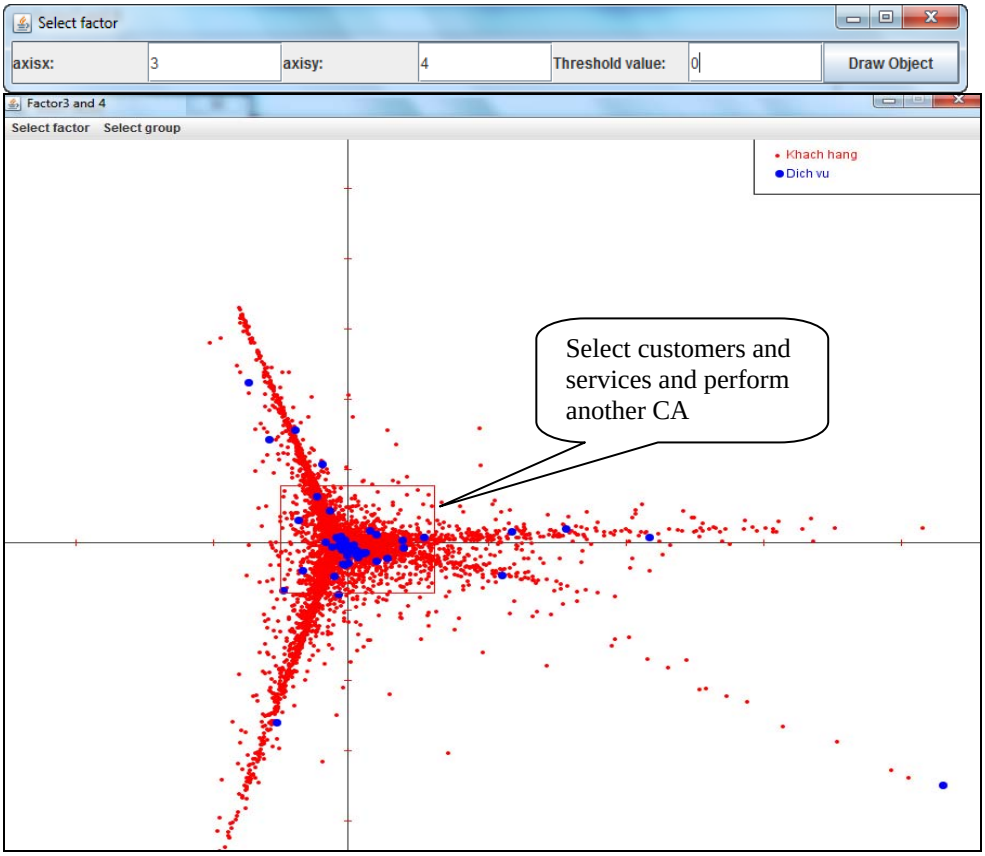

FIG. 5 – *Selecting customers and services for CA*

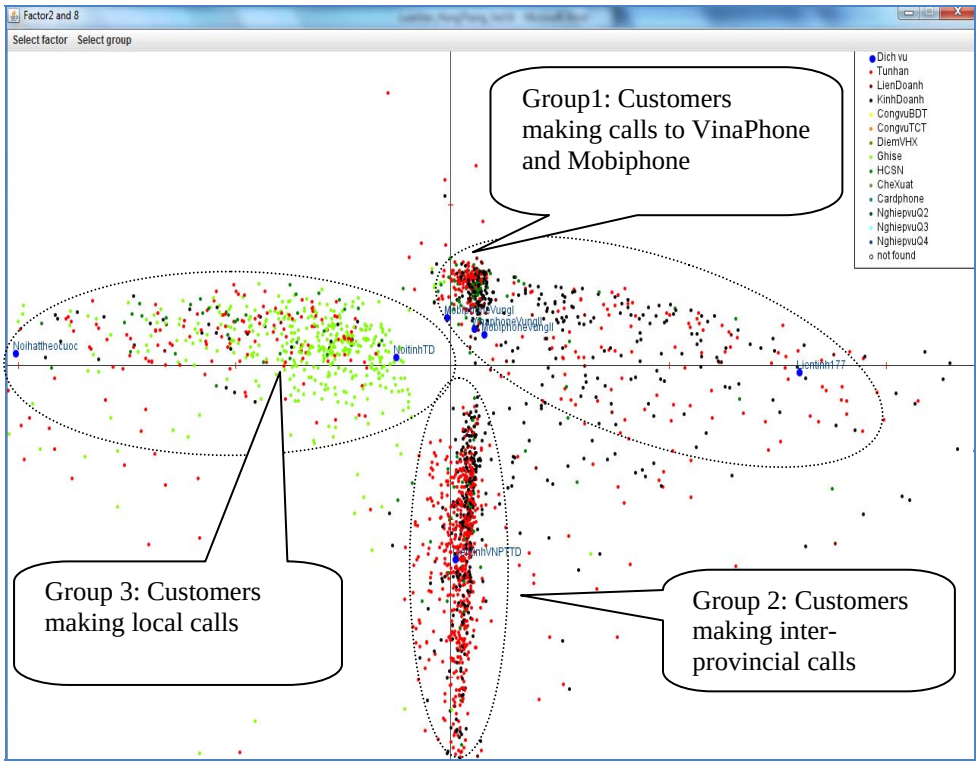

FIG. 6 – *Results of the second CA on the selected group of customers and services*

We found here some interesting information:

Group 1 and group 2 correspond to customers who make calls to VinaPhone, VNPT MobiPhone or use inter-provincial calls.

Group 3 consists of customers who use local and provincial call services.

4.3 "July to December 2008" Dataset

Analogous to the previous dataset, after the preprocessing step, we kept 7973 customers and 56 services. Customers and services are shown in Fig. 7. Three groups of customers characterized by associated service are found here. Group 1: customers using the 171 service (inter-province call); group 2: customers using the 177 service (inter-province call) and group 3: customers making local calls (within the province). By changing displayed axes, we found two other groups on the axes 15 and 16 (see Fig. 8). Group 1 consists of customers making call to phone numbers of Vinaphone (region II, these phone numbers belong to VNPT). Customers in group 2 often call numbers of Vinaphone (region I), Viettel (region I) and the inter-province EVN. These customers have some partners who subscribe to the Vinaphone, Viettel, and EVN.

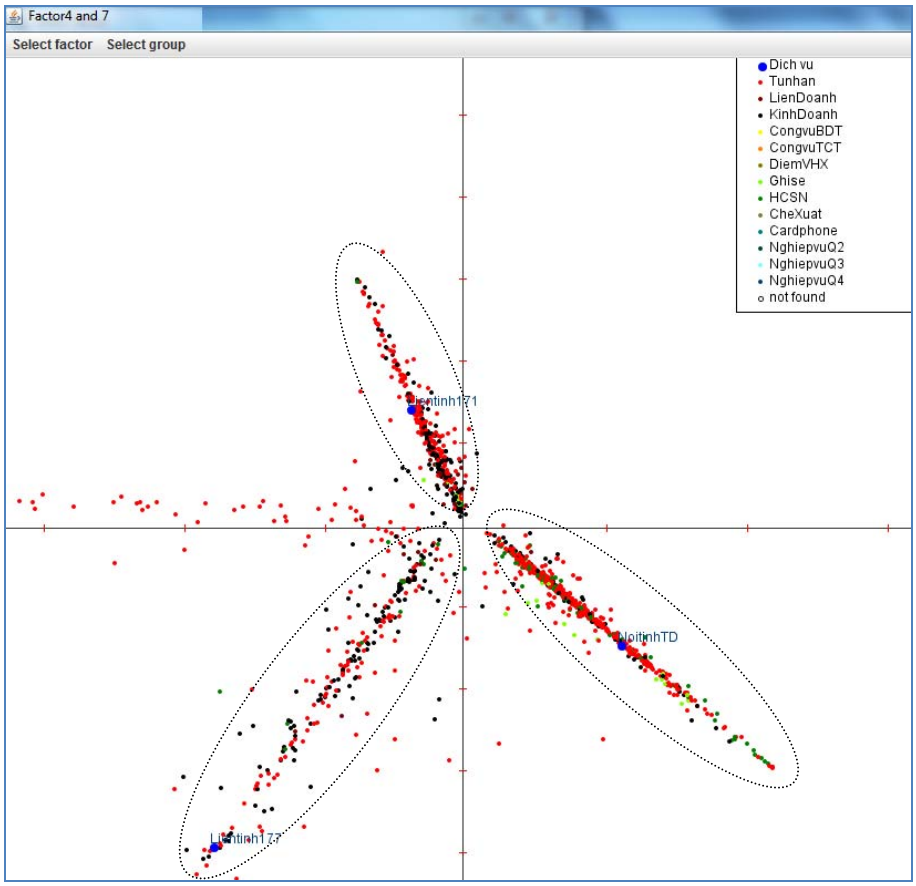

FIG. 7 – Results of CA on the second dataset

5 Conclusion and future works

We have presented in this paper the intensive use of Correspondence Analysis for exploration of Telecommunication Data. We have successfully adapted CA on Telecommunication Data and developed a visual tool which provides an interactive extraction of interesting information. The experimental results on two datasets have shown that groups of customers with same behavior could be found. This will help telecommunication companies to make appreciated policies for better caring their customers.

It is also worth if we monitor the behaviors of customers in time. We plan to integrate this feature in future works.

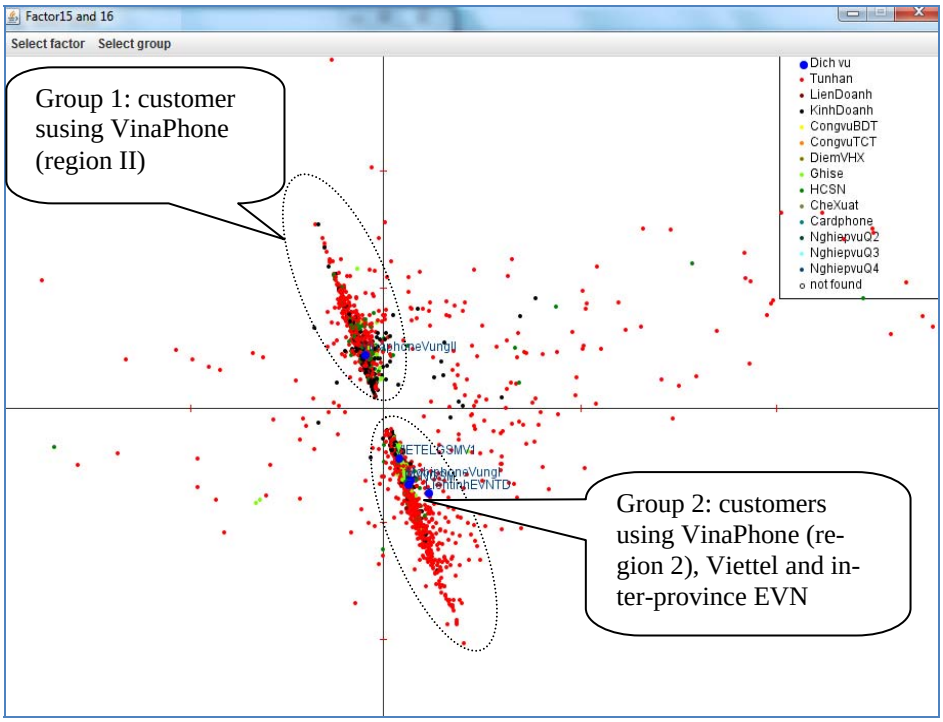

FIG. 8 – Two groups found in axes 15 and 16

References

Benzécri, J.-P. (1992), *Correspondence Analysis Handbook Statistics*, CRC Press.

Greenacre, M.J. (1984), *Theory and Application of correspondence analysis*. Academic Press, London.

Greenacre, M.J. (2007), *Correspondence analysis in practice, 2nd edition*. Chapman and Hall

Morin, A. (2004), "Intensive use of correspondence analysis for information retrieval," *26th International Conference on Information Technology Interfaces*, p. 255 – 258.

Pham, N-K, Morin A, Gros P, Le Q-T (2009a) "Intensive use of factorial correspondence analysis for large scale content-based image retrieval" in *Advances in Knowledge Discovery and Management*, AKDM'09, Springer-Verlag.

Pham, N-K, Morin A, Gros P, Le Q-T (2009b) "Accelerating image retrieval using factorial correspondence analysis on GPU", *in 13th International Conference on Computer Analysis of Images and Patterns, CAIP'09*, Lecture Notes in Computer science, Volume 5702, Pages 565-572, Münster, Allemagne.

Résumé

Nous présentons, dans cet article, un outil graphique interactif qui permet d'analyser, de visualiser et extraire des connaissances à partir des données de télécommunication en utilisant l'Analyse factorielle des correspondances (AFC). Les entreprises télécoms possèdent des très larges volumes des détails des appels (Call Record Details ou CRD). Les CRD indiquent quels clients ont utilisé quels services. L'AFC est une méthode originellement développée pour traiter des tableaux de contingence. Pour adapter l'AFC sur les CRD, nous les avons codé sous forme d'un tableau de contingence croissant les clients et les services proposés. Appliquer successivement l'AFC sur ce tableau nous permet d'explorer les groupes de clients qui ont la même habitude d'utilisation des services. Nous avons expérimenté la méthode proposée sur les données de l'entreprise Can Tho – Hau Giang Telecommunications. Les résultats ont montré l'intérêt de notre outil pour l'extraction des connaissances à partir des données de Télécommunication.

Identification et visualisation des partitions de réseaux sociaux à l'aide de points de vue sémantiques

Juan David Cruz[1], Cécile Bothorel[1], François Poulet[2]

[1]Département LUSSI, Télécom - Bretagne
Technopôle Brest-Iroise - CS 83818 - 29238 Brest Cedex 3
{juan.cruzgomez, cecile.bothorel}@telecom-bretagne.eu}
http://www.telecom-bretagne.eu/

[2]Université de Rennes 1 – IRISA
Campus de Beaulieu - 35042 Rennes cedex
francois.poulet@irisa.fr
http://www.irisa.fr/texmex

Résumé. Les algorithmes classiques de détection de communautés dans les réseaux sociaux utilisent l'information structurelle pour détecter des groupes, i.e. la topologie du graphe de relations. Toutefois, ils ne prennent en compte aucune information externe qui peut guider le processus et aider à la réalisation des analyses du réseau selon différentes perspectives. La méthode proposée utilise de façon conjointe, l'information sémantique du réseau social, représentée par des points de vue, et son information structurelle. Elle permet la combinaison entre les relations sociales explicites, les arêtes du graphe social, et les relations implicites, dites sémantiques, correspondant par exemple à des intérêts ou des usages similaires.

1 Introduction

Un réseau social est composé de groupes d'individus regroupés entre eux selon différents liens possibles. A l'intérieur d'un réseau social, il peut exister différents types de relations comme les amis, la famille, les collègues de travail... Les membres d'un réseau social peuvent aussi être décrits par leur appartenance à une entreprise ou des centres d'intérêts communs. Ce type d'information décrivant les relations et les individus composant le réseau social en définissent l'information sémantique.

On peut utiliser cette information sémantique pour analyser le réseau suivant différentes perspectives et pas en s'intéressant seulement à l'aspect structurel ou topologique du réseau. Nous présentons une méthode qui permet de combiner à la fois l'information topologique du réseau et l'information des membres du réseau (l'information sémantique). Cette information sémantique peut utilisée des sous-ensembles de l'information totale disponible, on parlera alors de points de vue. Un point de vue est un ensemble d'attributs qui représente une vision du réseau selon une perspective donnée. L'utilisation de cette information sémantique va

nous permettre de guider le processus de clustering et de changer la configuration de la partition en fonction du point de vue utilisé pour effectuer l'analyse du réseau social.

Le reste de cet article est organisé de la manière suivante : la section deux est un état de l'art sur les travaux de détection de communautés dans les réseaux sociaux, la section trois décrit la notion de points de vue dans les réseaux sociaux, la section 4 présente notre algorithme de clustering suivi de quelques résultats expérimentaux dans la section 5 avant la conclusion et les travaux futurs.

2 Détection de communautés dans les réseaux sociaux

De nombreuses méthodes ont été développées pour le clustering de graphe ou la détection de communautés dans les réseaux sociaux. De manière générale, ces méthodes sont vues comme un problème d'optimisation dont la fonction objectif est la maximisation d'un critère de qualité donné. Ces indices de qualité de la partition C sont basés sur le nombre d'arêtes à l'intérieur d'un groupe et entre les différents groupes. Nous allons présenter les différents indices de qualité des algorithmes classiques de partitionnement de graphes.

2.1 Indices de qualité du clustering

Un indice de qualité est une mesure du nombre de liens entre les communautés et du nombre de liens à l'intérieur de chaque communauté. L'idée est de trouver une configuration du graphe qui minimise le nombre de liens inter-cluster et maximise le nombre de liens intra-cluster. Etant donnée une partition $C \in A(G)$, avec A(G) l'ensemble de toutes les partitions possibles du graphe G, (Brades et al, 2008) définissent in indice générique pour la mesure de qualité d'une partition d'un graphe. Cet indice utilise deux fonctions indépendantes f et g qui mesurent respectivement la densité intra-cluster et l'eparsité inter-cluster. Cet index est défini de la manière suivante :

$$\text{index}(G) = (f(C)+g(C)) / N(G)$$

où N(G) est une fonction de normalisation qui vaut : $\max \{f(C')+g(C') : C' \in A(G)\}$.

En utilisant cette formulation (Gaetler, 2005) et (Brandes et al., 2008) définissent trois indices de qualité : le "coverage", qui mesure le poids de tous les liens intra-cluster par rapport au poids des tous les liens du graphe, la "conductance" qui est basée sur l'observation que si un cluster est très fortement connecté alors il faut éliminer un grand nombre de liens avant de le scinder en deux et la "performance" qui définit la qualité d'un cluster basée sur l'exactitude de la classification d'une paire de nœuds. Cette correction se base sur l'appartenance de deux nœuds connectés au même cluster ou de deux nœuds non connectés à deux clusters différents. En plus de ces critères, un quatrième a été introduit par (Newman et al., 2004) : Q, la "modularity" qui compare le nombre de liens à l'intérieur d'un cluster avec le nombre de liens entre les clusters, c'est-à-dire la densité de liens intra-cluster par rapport à

la sparsité de liens inter-clusters. Cet index est le plus couramment utilisé dans les méthodes de clustering (Fortunato, 2010).

2.2 Méthodes conventionnelles de clustering de graphes

Plusieurs méthodes ont été développées pour trouver les clusters d'un graphe. Ces méthodes cherchent à trouver les groupes minimisant le nombre de liens entre les groupes et maximisant le nombre de liens dans les groupes. Ces approches permettent de trouver la meilleure partition quand la matrice d'adjacence du graphe est creuse (Fortunato, 2010).

L'algorithme proposé par (Newman, 2001) cherche et élimine itérativement les liens ayant le plus grand dégré d'interposition. Ce processus permet de trouver les groupes qui sont faiblement connectés entre eux et dont les nœuds sont fortement connectés à l'intérieur d'un groupe. L'inconvénient majeur de leur approche est la complexité de calcul du degré d'interposition, la complexité générale de l'algorithme est en $O(mn^2)$ pour m arêtes et n nœuds, son coût est donc prohibitif pour des grands graphes.

Basé sur un algorithme de recherche en largeur d'abord qui permet de trouver les chemins les plus courts d'un nœud source s à tous les autres en O(m), (Brandes, 2001) et (Newman et al., 2004) ont proposé indépendamment une méthode en O(mn) permettant de trouver les nœuds "feuilles" et sommant les poids des arêtes de ses feuilles vers un nœud source. En répétant le processus pour tous les nœuds et sommant les scores, on obtient le degré d'interposition de toutes les arêtes du graphe.

Le "fast unfolding algorithm", proposé par (Blondel et al., 2008) est un algorithme d'agglomération pour trouver les communautés. Dans une première étape, chaque nœud est affecté à une communauté et la modularité initiale est calculée. Ensuite, chaque nœud *i* est enlevé de sa communauté pour être itérativement attribué à chaque communauté. A chaque affectation, le gain de la modularité est calculé et *i* sera attribué à la communauté donnant la plus grande valeur de gain. Si aucun gain positif n'est possible, *i* reste dans sa communauté initiale. Ce processus est appliqué jusqu'à ce qu'aucune amélioration de la modularité ne puisse plus être apportée.

Une fois cette première étape achevée, un nouveau graphe est construit, les nœuds sont les communautés trouvées à l'étape précédente et les arêtes sont la somme des poids des arêtes des communautés correspondantes. La première étape est alors de nouveau appliquée à ce nouveau graphe jusqu'à ce qu'aucune amélioration de la modularité ne puisse être apportée. La complexité de cet algorithme est linéaire en fonction des liens pour des graphes épars (Blondel et al., 2008), c'est-à-dire, avec une matrice d'adjacence contenant essentiellement des valeurs nulles.

(Du et al., 2007) proposent un algorithme pour la détection de communautés dans les réseaux sociaux de grandes tailles. Leur méthode est basée sur l'énumération de toutes les cliques maximales, i.e. les sous-graphes complets non contenus dans un autre sous-graphe complet. Une fois l'ensemble de ces cliques énumérées, ils génèrent des noyaux associés à ces cliques et calculent la détection de communautés en affectant les nœuds à chaque noyau : chaque noyau est une communauté et chaque nœud est assigné à la communauté la plus proche. Ensuite, ils essayent d'optimiser la modularité en cherchant quelle fusion d'une paire de communautés améliore la modularité. Les deux communautés qui seront fusionnées sont celles qui produisent la valeur la plus haute de modularité. Cet algorithme a une complexité

en O ($D * M_C * Tri^2$) où D est le degré maximal de l'ensemble des nœuds, M_C est la taille de la plus grande clique et Tri est le nombre de triangles présents dans le graphe.

La plupart des algorithmes classiques vont trouver des partitions disjointes. Cependant beaucoup de réseaux sociaux réels ont des acteurs qui peuvent appartenir à différentes communautés, donc des communautés qui se recouvrent. (Pizzuti, 2009) présente une méthode permettant de détecter des communautés recouvrantes. Sa méthode utilise un algorithme génétique et la qualité d'un individu (fitness) est calculée de la manière suivante :

$$CS = \sum_{i \in S} score(Si)$$

où S représente l'ensemble des communautés et score (Si) est la mesure de qualité qui compare les arêtes dans le groupe i et les arêtes du groupe vers le reste du réseau.

(Lipczak et al., 2009) proposent aussi un algorithme génétique pour la détection de communautés. Dans leur cas, les individus sont représentés par une chaîne de caractères des groupes, soit un vecteur de taille n contenant le nombre de nœuds des n groupes. Pendant les phases de sélection et crossover, les gènes sont sélectionnés en fonction des améliorations potentielles qu'ils peuvent apporter à la fonction fitness. Cette fonction est composée de trois mesures : la "normalized cut" proposée par (Shi et al., 2000), la "modularity" proposée par (Newman et al., 2004) et la "silhouette width") proposée par (Rousseeuw, 1987).

D'autres méthodes de clustering, comme le "Markov Clustering", l'"Iterative Conductance Cutting" et le "geometric minimum spanning tree" sont comparées dans (Brandes et al., 2008) et des méthodes d'évaluation de communautés dans (Kwak et al., 2009) et (Günter et al., 2003).

En général, pour la découverte de communautés, ces méthodes ne prennent en compte que la structure du graphe et n'incluent aucune autre information associée aux nœuds.

3 Définition de points de vue dans les réseaux sociaux

Les réseaux socio-sémantiques contiennent une part importante d'information qui ne vient pas de leur topologie, mais des différents contextes de ces réseaux. Cette information peut être associée aux acteurs et à leurs relations au sein du réseau et peut fournir plus d'éléments pour l'analyse du réseau selon différentes perspectives.

Par exemple, il est possible pour les acteurs d'un réseau social d'utiliser les informations sur les groupes auxquels ils appartiennent tels que les associations, entreprises ou leurs localisations géographiques.

Avec ce genre d'informations, il est possible de définir deux ensembles de caractéristiques, un pour l'ensemble des acteurs et un pour les relations ce qui va permettre d'effectuer une analyse du réseau, non seulement d'un point vue structurel mais aussi d'un point de vue sémantique.

Il est aussi possible de n'utiliser qu'un sous-ensemble des caractéristiques sémantiques, ce qui permet aussi l'analyse selon un point de vue particulier. Cette approche permet l'extraction de connaissances en changeant les points de vue ou en définissant des filtres sur plusieurs variables de l'ensemble des caractéristiques de l'ensemble de données.

Nous allons détailler la représentation des points de vue dans les sections suivantes.

3.1 Quelques notations

Un réseau social peut être représenté par un graphe non orienté *G(V,E)* où *V* est l'ensemble des sommets (représentant les acteurs du graphe) et *E* l'ensemble des arêtes représentant les relations entre les acteurs.

Si v_i et v_j sont deux sommets de V et e(x,y) l'arête définie par les sommets x et y alors $e(v_i,v_j) \in E$ si v_i et v_j sont voisins. Comme le graphe est non orienté, $e(x,y) \equiv e(y,x)$, $\forall$ (x,y) $\in$ V et e (x, x) $\notin$ E, $\forall x \ni V$.

Etant donné un graphe G, C={C1, C2, ..., Ck} est une partition de V en k sous-ensembles disjoints Ci.

Soit F_V l'ensemble des attributs des acteurs du réseau social, il peut être représenté par une matrice de taille $|V| \times |F_V|$ et soit l'ensemble F_E des attributs associés à chaque arête, il peut être représenté par une matrice de taille $|E| \times |F_E|$.

Etant donné un graphe G(V,E) et un ensemble d'attributs, un réseau socio-sémantique S peut être défini comme le tuple :

$$S = < G, F_V , F_E > \quad (1)$$

A partir de ces notations, nous allons définir ce qu'est un point de vue.

3.2 Représentation d'un point de vue

Nous allons définir un point de vue en utilisant les attributs des nœuds. Etant donné un réseau sémantique $S = <G, F_V , F_E >$, soit $F_V^* \ni P(F_V) \setminus F_V$, où P(A) est l'ensemble des partitions de A, l'ensemble des attributs utilisés pour définir le point de vue PoV.

Pour chaque sommet $v_i \ni V$, il y a un vecteur u_i de taille $|F_V^*| = f$. Si le sommet i a l'attribut p, $1 \leq p \leq f$ dans F_V^*, alors $u_i = 1$ et 0 sinon. Donc chaque vecteur u peut être défini comme suit :

$$u_i = v_i \times F_V^* \quad (2)$$

avec $v_i \ni V$.

Ensuite un point de vue est défini comme l'ensemble de toutes les instances de l'ensemble F_V^*, soit d'après (2) :

$$PoV_{Fv^*} = \bigcup_{i=1}^{|V|} u_i \quad (3)$$

Le tableau 1 montre un exemple où l'ensemble des nœuds reçoivent une instance des attributs de u, on peut remarquer que différents nœuds peuvent avoir les mêmes instances de u.

	Point de vue			
Noeud	Attribut 1	Attribut 2	...	Attribut f
1	1	0	...	0
2	0	1	...	1
...	...	...	...	...
n	1	0	...	1

TAB. 1 – *Exemple d'assignation des attributs aux nœuds du réseau*

Il est aussi possible de définir un point de vue en utilisant l'information des arêtes à la place de celle des nœuds. Dans ce cas, chaque arête comporte les informations des nœuds qu'elle relie et son poids. Elle peut de plus contenir des informations sur le type de relation (par exemple ami ou famille).

Au lieu d'utiliser seulement l'information des arêtes, on peut ajouter les points de vue des nœuds pour obtenir ainsi une perspective enrichie d'information sémantique.

4 Utilisation des points de vue dans le clustering

Le but est d'utiliser simultanément l'information structurelle du graphe et l'information sémantique liée aux membres du réseau social, comme leurs relations.

En utilisant ces types d'information, il est possible de guider le processus de clustering de graphe en ajoutant l'information sur la similarité des nœuds tirée du contexte réel. Pour ce faire l'algorithme de détection de communautés est divisé en deux étapes. Pendant la première, les points de vue sont traités à l'aide d'un algorithme de cartes de Kohonen (Kohonen, 1997) ou cartes auto-organisatrices pour obtenir des groupes basés sur la similarité des attributs des nœuds du graphe. Les groupes obtenus dans cette première étape sont ensuite utilisés pour modifier le poids des arêtes du graphe et ensuite un algorithme classique de détection de communautés dans les graphes est appliqué dans une deuxième étape.

L'utilisation de plusieurs étapes dans l'algorithme de clustering est courante dans les algorithmes d'apprentissage. L'utilisation d'une phase de pré-traitement permet d'améliorer la qualité des résultats de certains algorithmes. Par exemple (Gonzales et al, 2006) utilisent des cartes auto-organisatrices pour discriminer les individus normaux ou anormaux dans un système artificiel de défense immunitaire, (Deng et al., 2007) présentent une technique qui utilise d'abord un algorithme de clustering hybride pour créer un réseau de neurones flou et utilisent ensuite ce réseau pour prédire les émissions de particules diesel.

4.1 Première phase : clustering sémantique

Etant donné un point de vue dérivé d'un ensemble F_V* comme décrit dans la section 3, chaque nœud peut être caractérisé par son vecteur d'attributs ou une instance *u* du point de vue. Il est possible d'utiliser ses vecteurs en entrée d'un algorithme de classification non supervisée comme les cartes auto-organisatrices (Kohonen, 1997). Cela permet de créer des groupes de nœuds suivant la similarité de leurs attributs, c'est-à-dire les instances de *u* sont les données en entrée de l'algorithme.

L'algorithme de cartes de Kohonen utilisé a un réseau N basé sur une grille rectangulaire de taille f x f neurones, avec $f = |F_V *|$ le nombre d'attributs utilisés dans le point de vue. Les valeurs initiales des poids sont tirées aléatoirement. Les poids des neurones sont ajustés selon leur proximité au neurone gagnant. Un taux d'apprentissage η est utilisé pour éviter les maxima locaux et des convergences prématurées. Après chaque itération le taux d'apprentissage est réduit par un factor de $0 < \varepsilon < 1$. Le voisinage est calculé avec une taille t et le neurone gagnant de centre c.

La complexité de l'algorithme est proportionnelle au nombre d'attributs du point de vue utilisé, au nombre de nœuds et la taille du réseau de neurones. Elle peut s'exprimer ainsi :

$$T = O(f^3.n) \quad (4)$$

avec *n* le nombre de nœuds du graphe et *f* le nombre d'attributs utilisés dans le point de vue.

En sortie de l'algorithme, on obtient une partition C_{SOM} des nœuds assignés aux neurones.

4.2 Deuxième phase : clustering structurel et détection de communautés

Une fois la partition sémantique C_{SOM} calculée on peut alors entrer dans la seconde phase de la méthode. Dans cette étape, on utilise un algorithme classique de détection de communautés, le "fast unfolding algorithm", proposé par (Blondel et al., 2008) et présenté dans la section 2. Cet algorithme utilise la modularité *Q*, présentée par (Newman et al., 2004), comme mesure de qualité.

Avant l'exécution du "fast unfolding algorithm", on inclue les informations obtenues lors de la première phase. Cela est effectué par le changement des poids des arêtes en fonction de la partition obtenue C_{SOM}. Pour chaque paire de sommets $v_i, v_j \ni V, \forall\ v_i \neq v_j$, le poids de l'arête $e(v_i, v_j)$ est modifiée par la distance euclidienne des instances du point de vue PoV correspondant à chaque nœud :

$$w_{ij} = 1 + \alpha\,(1 - d\,(Nij))\,\delta ij \quad (5)$$

avec $\alpha \geq 1$ une constante, d (Nij) est la distance entre les neurones i et j, et $\delta ij = 1$ si v_i et v_j appartiennent au même cluster dans C_{SOM} et 0 sinon. L'algorithme 1 montre comment l'information sémantique est utilisée dans le graphe G.

Une fois que les poids sont modifiés selon l'équation 5, une partition, C_{SOMFU} est calculée en utilisant le "fast unfolding algorithm". Cette nouvelle partition contient l'ensemble des communautés finales et l'information structurelle.

Comme notre approche ajoute un pré-traitement pour trouver une partition sémantique, C_{SOM} la complexité globale de l'algorithme est fonction de la complexité du clustering sémantique et de celle de l'algorithme de fast unfolding.

La complexité de l'étape de clustering sémantique est donnée par l'équation 4 et celle de l'algorithme de fast unfolding est linéaire en nombres de nœuds dans le cas de matrices d'adjacences creuses selon (Blondel et al., 2008).

Donc, comme les matrices des réseaux sociaux sont en général creuses (Bouklit, 2006), la complexité globale de notre algorithme est en $O(|F_V|^3\ |V|)$, avec V le nombre de nœuds et F_V le nombre d'attributs sémantiques utilisés.

5 Premières expérimentations

Nos premières expérimentations ont été réalisées sur trois graphes générés artificiellement. Le tableau 2 décrit les caractéristiques des graphes générés, la mesure de densité utilisée est :

$$\delta = \frac{2x|E|}{|V|x(|V|-1)}$$

avec |E| le nombre d'arêtes du graphe et |V| son nombre de sommets. Comme le nombre maximal d'arêtes d'un graphe de n = |V| nœuds est n x (n-1) /2, la valeur maximale de la densité est 1.

Graphe	Noeuds	Arêtes	Densité	Modularité	PoV	Attributs
1	200	3892	0.2	-5.1216×10^{-3}	1	5
2	5389	27347	1.8836×10^{-3}	-2.5192×10^{-3}	2	33
					3	4

TAB. 2 – *Description des caractéristiques des graphes utilisés pendant l'expérimentation. Le graphe 1 est généré artificiellement et le graphe 2 est tiré de Twitter. Le point de vue 1 est généré aléatoirement et les ponts de vues 2 et 3 sont construits à partir des données de Twitter.*

Le graphe 1 a été généré artificiellement pour tester la méthode de manière générique. Le graphe 2 est un exemple réel tiré de Twitter. On peut remarquer que la densité du graphe artificiel est plus élevée que celles des graphes réels.

Pour notre expérimentation, nous avons comparé les résultats de notre approche avec deux algorithmes classiques de clustering. Le premier est l'algorithme de cartes auto-organisatrices qui calcule une partition C_{SOM} basée uniquement sur l'information sémantique, le second est l'algorithme de fast unfolding qui calcule les clusters C_{FU} en se basant uniquement sur l'information structurelle et le résultat de notre approche $C_{SOM\text{-}FU}$.

Pour mesurer la qualité des résultats, nous avons utilisé la distance euclidienne moyenne des nœuds au sein d'un cluster et la modularité *Q* pour évaluer la qualité de la partition d'un point de vue structurel.

Les points de vue utilisés dans nos expérimentations sont les suivants :

1) Point de vue aléatoire : on génère un point de vue avec cinq attributs choisis aléatoirement. Il y a 2^5 possibilités de même probabilité, une même instance peut être attribuée à plusieurs nœuds.

2) Le fuseau horaire : il représente l'une des 33 valeurs enregistrées dans l'ensemble de données et est le décalage en seconde avec le méridien de référence. Chaque instance décrit la présence d'amis dans chaque fuseau horaire.

3) Le profil utilisateur : le premier attribut indique si l'utilisateur a plus d'amis que de suiveurs. Les utilisateurs qui ont plus de suiveurs que d'amis sont en général des personnes ou des organisations dont les messages intéressent un grand nombre de personnes (comme les hommes politiques ou les célébrités). Les trois autres attributs visent à décrire le comportement de l'utilisateur en ce qui concerne le nombre de messages envoyés. Les attributs sont : en dessous de la moyenne, entre la moyenne et la moyenne plus trois fois l'écart-type et supérieur à la moyenne plus trois fois l'écart-type. Dans l'ensemble de données, environ 82% des utilisateurs sont en dessous de la moyenne des messages envoyés.

5.1 Résultats sur le graphe 1 avec le point de vue 1

La première expérience a été réalisée sur le graphe de 200 nœuds et 3892 arêtes. Dans ce cas, les groupes obtenus par l'algorithme de fast unfolding ont une distance intracluster plus élevée. Notons que plus la distance entre deux nœuds est importante, plus la dissimilarité des attributs du point de vue est élevée.

Clustering	Q	Distance intracluster	Ecart-type
C_{SOM}	-0.0056	0.2793	0.2370
C_{FU}	0.1885	1.5375	0.4117
C_{SOM-FU}	0.5389	1.0833	0.5692

TAB. 3 – *Comparaison des résultats des algorithmes classiques et de notre approche sur le graphe 1.*

Les résultats présentés dans le tableau 3 montrent qu'avec notre approche la distance moyenne entre les nœuds au sein d'un cluster est inférieure à celle obtenue en utilisant que l'information structurelle du graphe.

Cela signifie que les individus des groupes obtenus, en prenant aussi en compte l'information sémantique dans notre approche sont plus similaires que ceux obtenus sans cette information sémantique. De plus, la modularité de la partition est plus élevée à cause du changement de la valeur des poids en fonction des groupes sémantiques, la partition obtenue est donc de meilleure qualité.

Les résultats de la première ligne du tableau sont ceux obtenus par l'algorithme de cartes auto-organisatrices. Dans ce cas la modularité est moins bonne que dans le cas de l'algorithme sur le graphe brut, mais la distance moyenne est plus faible, comme nous nous y attendions.

Ces résultats montrent que les groupes obtenus en utilisant les points de vue peuvent influencer les résultats des algorithmes de clustering classiques. La modularité a été améliorée de manière significative tout en préservant la similarité intra-cluster.

5.2 Résultats sur le graphe 2 avec le point de vue 2

Nous présentons les résultats obtenus avec un graphe de 5389 nœuds et 27347 arêtes tiré d'un ensemble de données de Twitter (composé de 204000 nœuds et 326000 arêtes).

Clustering	Q	Distance intracluster	Ecart-type
C_{SOM}	-0.0075	0.3697	0.1059
C_{FU}	0.5728	1.8091	1.3584
C_{SOM-FU}	0.5747	1.1947	0.8489

TAB. 4 – *Comparaison des résultats des algorithmes classiques et de notre approche sur le graphe 2 avec le point de vue 2.*

Comme on peut le voir dans les résultats présentés dans le tableau 4, la distance intracluster de notre approche est inférieure à celle obtenue par l'algorithme utilisant seulement l'information structurelle du graphe. La modularité obtenue dans les deux cas est presque la même. Ceci est dû à la composition du point de vue qui intègre des informations sur la localisation des amis. On peut penser que la localisation des amis et la liste des amis sont des informations très similaires.

Les résultats suivants concernent un point de vue différent qui n'est pas lié aux relations personnelles dans le réseau mais à leur type d'interaction dans le réseau.

5.3 Résultats sur le graphe 2 avec le point de vue 3

Ces résultats concernent le même graphe 2 que dans le paragraphe précédent, mais avec un point de vue différent. Les résultats sont présentés dans le tableau 5.

Clustering	Q	Distance intracluster	Ecart-type
C_{SOM}	-0.2991	0	0
C_{FU}	0.5728	0.7100	0.6565
C_{SOM-FU}	0.6351	0.5507	0.5577

TAB. 5 – *Comparaison des résultats des algorithmes classiques et de notre approche sur le graphe 2 avec le point de vue 3.*

Le point de vue utilisé ici comporte deux attributs : le premier indique si l'utilisateur a plus d'amis que de suiveurs ou non et le second est une variable catégorique à trois valeurs transformée en trois variables booléennes. Le nombre de choix possibles est donc de 2x3=6 possibilités différentes.

Le résultat des cartes auto-organisatrices montrent que le clustering en six groupes chacun reflétant l'une des possibilités décrites est la bonne classification, ce qui explique la distance obtenue.

L'utilisation de l'information sémantique avec les cartes auto-organisatrices permet d'obtenir un clustering de meilleure qualité qu'avec le graphe, cependant la modularité est moins bonne qu'avec le graphe, cela montre que les clusters obtenus par l'algorithme de cartes auto-organisatrices sont totalement indépendant de la structure du graphe.

Dans le cas du clustering utilisant l'information de structure et l'information sémantique, les résultats sont meilleurs à la fois en ce qui concerne la modularité et la distance intracluster.

6 Conclusion et perspectives

L'information contenue dans les réseaux socio-sémantiques est liée à la structure du graphe et à certains attributs des individus. Une telle information permet d'analyser le réseau sous différents points de vue.

Les algorithmes classiques de détection de communauté utilisent seulement l'information de la structure du graphe et ne prennent pas en compte l'information sémantique, qui pourrait être utilisée pour améliorer le processus de clustering.

En pondérant les arêtes par des poids obtenus par clustering sémantique, l'information sémantique du réseau est utilisée dans le processus de détection de communautés et les deux types d'information (structurelle et sémantique) sont mixées pour obtenir et visualiser le réseau social selon le point de vue voulu.

En ce qui concerne le temps d'exécution de notre méthode, la complexité est plus importante que les méthodes basées sur les graphes. Aujourd'hui, cela restreint le nombre d'attributs sémantiques que l'on peut utiliser avec les cartes auto-organisatrices.

Un grand nombre de dimensions peut poser des problèmes à l'algorithme de cartes auto-organisatrice à cause de la malédiction de la dimension (Bellman, 1957) et de comment est mesurée la distance sémantique. Nous étudierons donc les propriétés statistiques des points de vue pour essayer de pallier cet effet.

Nous allons aussi continuer à étudier l'influence du point de vue dans le processus de détection de communautés en incluant cette information au niveau des arêtes. Nous étudierons également comment visualiser efficacement un réseau social hiérarchique.

Références

Bellman R.E., *Dynamic programming*, Princeton University Press, 1957.

Blondel V. D., Guillaume J.-L., Lambiotte R., Lefebvre E., Fast unfolding of communities in large networks, *Journal of Statistical Mechanics : Theory and Experiment*, vol. 2008, n° 10, p. P10008 (12pp), 2008.

Bouklit M., Autour du graphe duWeb : Modélisations Probabilistes de l'Internaute et Détection de structures de Communauté, PhD thesis, Université Montpellier II, 2006.

Brandes U., A Faster Algorithm for Betweenness Centrality, *Journal of Mathematical Sociology*, vol. 25, p. 163-177, 2001.

Brandes U., Gaetler M., Wagner D., Engineering graph clustering : Models and experimental evaluation, *Journal of Experimental Algorithmics*, vol. 12, p. 1-26, 2008.

Deng J., Stobart R., Plianos A., Combined hybrid clustering techniques and neural fuzzy networks to predict diesel engine emissions, *Proceedings of The International Conference on Systems, Man and Cybernetics – ISIC*, p. 3609 -3614, oct., 2007.

Du N.,Wu B., Pei X.,Wang B., Xu L., Community detection in large-scale social networks, *WebKDD/SNA-KDD '07 : Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, ACM, New York, NY, USA, p. 16-25, 2007.

Fortunato S., Community detection in graphs, *Physics Reports*, vol. 486, n° 3-5, p. 75 - 174, 2010.

Gaetler M., *Network Analysis : Methodological Foundations*, Springer Berlin / Heidelberg, chapter Clustering, p. 178 - 215, 2005.

González F. A., Galeano J. C., Rojas D. A., Veloza-Suan A., A neuro-immune model for discriminating and visualizing anomalies, *Natural Computing*, vol. 5, p. 285-304, 2006.

Günter S., Bunke H., Validation indices for graph clustering, *Pattern Recognition Letters*, vol. 24, n° 8, p. 1107-1113, 2003.

Kohonen T., *Self-Organizing Maps*, Springer, 1997.

Kwak H., Choi Y., Eom Y.-H., Jeong H., Moon S., Mining communities in networks : a solution for consistency and its evaluation, *IMC '09 : Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, ACM, New York, NY, USA, p. 301-314, 2009.

Lipczak M., Milios E., Agglomerative genetic algorithm for clustering in social networks, *GECCO '09 : Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, ACM, New York, NY, USA, p. 1243-1250, 2009.

Newman M. E., Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality., *Physical Review. E, Statistical Nonliner and Soft Matter Physics*, vol. 64, p. 7, July, 2001.

Newman M. E. J., Girvan M., Finding and evaluating community structure in networks, *Physical Review. E, Statistical Nonliner and Soft Matter Physics*, Feb, 2004.

Pizzuti C., Overlapped community detection in complex networks, *GECCO '09 : Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, ACM, New York, NY, USA, p. 859-866, 2009.

Rousseeuw P., Silhouettes : a graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics*, vol. 20, n° 1, p. 53-65, 1987.

Shi J., Malik J., Normalized Cuts and Image Segmentation, *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 22, n° 8, p. 888-905, 2000.

Summary

Classic algorithms for community detection in social networks use the structural information to identify groups in the social network, i.e., how clusters are formed according to the topology of the relationships. However, these methods don't take into account any semantic information which could guide the clustering process, and which may add elements to do further analyses. The method we propose, uses in a conjoint way, the semantic information from the social network, represented by the points of view, and its structural information. This information integrates the relationships, expressed by the edges on one hand, and the implicit relations deduced from the semantic information on the other hand.

Détection visuelle d'événements dans des grands réseaux d'interaction dynamiques. Application à l'Internet

Bénédicte Le Grand, Matthieu Latapy

LIP6, Université Pierre et Marie Curie - Paris VI
4 place Jussieu, 75252 Paris, France
{benedicte.le-grand, matthieu.latapy}@lip6.fr

Résumé. L'objectif des travaux présentés dans ce papier est de faciliter la détection visuelle d'événements dans des réseaux d'interaction dynamiques de grande taille.
Deux méthodes de visualisation classiques et «exhaustives» ont été étudiées, qui repré-sentent l'évolution des liens du réseau au fil du temps. Les limites liées au facteur d'échelle nous ont conduits à proposer deux métaphores restreintes au suivi des nœuds du réseau. Les forces, les limites et la complémentarité de ces quatre métaphores nous ont permis de déga-ger une ébauche de méthodologie de détection d'événements dans la dynamique de grands réseaux d'interaction.
Les visualisations et la méthodologie présentées dans cet article sont génériques et appli-cables à tout type de nœuds et de liens ; elles sont ici appliquées pour illustration à un sous-ensemble du réseau Internet.

1 Contexte et objectifs

Alors que l'analyse et la visualisation de grands réseaux sont au cœur de nombreux travaux, l'étude de la dynamique de ces réseaux soulève encore de nombreux défis. L'objectif de nos travaux est de faciliter la détection visuelle d'événements (par exemple d'anomalies) liés à la dynamique de réseaux de grande taille. Bien que les métaphores et la méthodologie de visualisation présentées dans cet article soient génériques et applicables à tout type de nœuds et de liens, elles sont ici illustrées sur un cas particulier : un sous-ensemble du réseau Internet. Dans ce contexte, les nœuds du réseau sont des machines et les liens entre les nœuds correspondent aux liens physiques entre ces machines. Les détails de la construction du réseau sont présentés dans la Section suivante.

1.1 Réseau d'étude

Le réseau étudié ici est constitué des nœuds et des liens traversés au fil du temps par une machine donnée (désignée comme la *source* dans la suite de cet article) pour atteindre 3000

destinations de l'Internet[1]. Pour connaître, à un instant donné, le chemin suivi par les paquets pour aller de la source vers la première destination, le processus est le suivant :

- La source émet un message (plus spécifiquement un *paquet IP*) vers la première destination en limitant à 1 le nombre de routeurs qu'il est autorisé à traverser (grâce au champ *Time To Live* de l'en-tête IP). Le premier routeur traversé, puisqu'il n'est pas autorisé à transmettre le paquet à un autre routeur, envoie alors un message d'erreur à la source, pour lui signifier que le paquet n'a pas pu arriver à destination ; la source connait ainsi l'adresse du premier routeur traversé (qui se trouve dans le champ *adresse source* de l'en-tête IP du paquet).
- On recommence l'opération en limitant successivement le champ *Time To Live* à 2, 3, …N ; on obtient ainsi l'adresse des 2e, 3e, … N-ième routeurs traversés par les paquets envoyés par la source vers la première destination.

Ce processus est ensuite répété pour chacune des destinations visées (3000 destinations distinctes), permettant d'obtenir une carte du réseau à partir de la source, comme l'illustre la Figure 1.

Cette carte a été recalculée périodiquement, toutes les 15 minutes environ pendant quasiment 3 semaines, 2000 fois en tout (voir (Magnien et al., 2009) pour plus de détails sur la collecte des données et sur l'optimisation du processus pour limiter le nombre de messages à envoyer). Les données étudiées dans ce papier représentent donc 2000 *itérations*, aussi appelées *passes*, qui constituent autant de vues successives de l'évolution de la topologie du réseau au fil du temps.

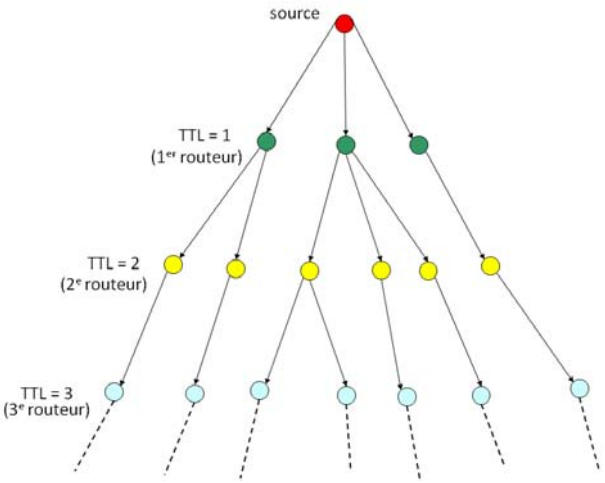

FIG. 1 – *Construction du réseau*

[1] Ces machines destinations appartiennent à la plate-forme *PlanetLab* et sont réparties partout dans l'Internet.

1.2 Objectif

L'objectif des travaux présentés dans cet article est de détecter visuellement des évolutions inattendues du réseau : de tels événements sont intrinsèquement liés à la *dynamique* du réseau, dans la mesure où la présence d'un nœud ou d'un lien sur une carte donnée n'est jamais anormale en soi ; c'est l'apparition ou la disparition de nœuds ou de liens qui peut (éventuellement) constituer des événements anormaux dans l'évolution d'un réseau IP. Des modifications soudaines et ponctuelles des routes suivies par les messages peuvent refléter des anomalies, alors que l'alternance entre plusieurs chemins est parfois normale dans le cadre d'un équilibrage de la charge sur le réseau (appelé *load balancing*).

Une première difficulté consiste donc à définir la notion d'événement ; les travaux de (Hamzaoui et al., 2010), appliqués au même type de données, visent à détecter automatiquement les événements « statistiquement significatifs ». Une interprétation est ensuite nécessaire pour identifier ceux qui constituent effectivement des anomalies. Pour cela, les instants particuliers détectés sont étudiés et comparés à des notifications d'erreurs dans le réseau (tickets d'Abilène). Les statistiques peuvent s'avérer peu intuitives pour l'utilisateur final ; par ailleurs, l'intérêt du data mining visuel ayant été prouvé dans de nombreux contextes, l'objectif de ce papier est d'étudier comment la visualisation pourrait faciliter la détection d'événements dans la dynamique de réseaux de grande taille.

Les auteurs de (Teoh et al., 2003) utilisent des métaphores de représentation classiques pour détecter des anomalies de routage du protocole BGP. L'interactivité permet aux utilisateurs d'étudier en détail certains systèmes autonomes (AS) pour interpréter des anomalies qui se sont produites à des instants connus (ce qui n'est pas applicable à nos travaux, puisque l'on ignore quand des événements se sont produits). Ces travaux ont été étendus dans (Teoh et al., 2006), où les visualisations sont couplées à du data mining et permettent d'obtenir des résultats intéressants dans ce contexte spécifique. Dans nos travaux, nous tentons de conserver une approche aussi générique que possible, afin de pouvoir l'appliquer à tout type de réseaux d'interaction (par exemple des réseaux sociaux).

De nombreuses visualisations de réseau (en particulier des tentatives de représentation du réseau Internet) ont été proposées – souvent par des opérateurs ou des fournisseurs d'accès (Akamai), mais leur intérêt est le plus souvent de déceler des tendances globales, car les événements exceptionnels sont masqués par la très grande quantité de données (voir (Claffy et al.) qui décrit *Mapnet*, un outil de visualisation de l'infrastructure des backbones de fournisseurs internationaux). Par ailleurs, il s'agit le plus souvent de représentations statiques qui n'ont pas vocation à être mises à jour fréquemment.

D'autres types de réseaux d'interaction ont récemment suscité un grand intérêt en terme de visualisation ; c'est en particulier le cas de la blogosphère (Discovery, 2007).

A une échelle plus réduite que l'Internet, ce qui est également le cas du réseau étudié ici, des outils de visualisation de graphes tels que *GUESS* (Guess), *Gephi* (Gephi), *Tulip* (Tulip), *Pajek* (Batageli, 2009) ou *Commetrix* (Commetrix), sont extrêmement intéressants. Ces outils permettent en effet de générer des représentations visuelles de qualité (la représentation

proposée en section 2.1 utilise d'ailleurs l'outil *GUESS*). Notre objectif n'est pas de développer un nouvel outil, mais une méthodologie pouvant s'appuyer sur des outils existants.

Notre approche consiste donc à étudier des métaphores « naturelles » de visualisation de graphes et à proposer d'autres représentations du réseau. Certaines représentations sont intrinsèquement statiques et leurs différentes instances doivent être comparées les unes aux autres pour retrouver une dynamique (on peut par exemple visionner des cartes successives sous forme de vidéo), alors que d'autres reflètent la dynamique –partielle ou globale-sur une image unique.

La suite de l'article est organisée comme suit : nous présentons tout d'abord (Section 2) deux méthodes de visualisation classiques et «exhaustives», représentant les liens du réseau (et, pour l'une d'entre elle, les liens et les nœuds). Les limites liées au facteur d'échelle nous conduisent à proposer dans la Section 3 deux métaphores originales et restreintes aux seuls nœuds du réseau. Les forces, les limites et la complémentarité de ces quatre métaphores nous permettent de conclure en proposant une ébauche de méthodologie de détection d'événements dans la dynamique de grands réseaux d'interaction ; nous concluons ce papier en présentant les perspectives d'avenir de ces travaux.

2 Visualisation des liens du réseau

2.1 Représentation sous forme de graphe (d'arbre)

La visualisation d'un réseau d'interaction sous forme de graphe, constitué des machines du réseau et des liens physiques entre elles, est très naturelle ; de plus, elle constitue une représentation très complète, puisqu'elle permet de visualiser aussi bien les liens que les nœuds du réseau. L'étude des liens permet de détecter un plus grand nombre d'événements que le seul examen des nœuds : en effet, les liens sont forcément modifiés en cas d'apparition ou de disparition de nœuds, mais ils peuvent également évoluer au sein du même ensemble de nœuds (ce qui est indétectable si l'on se limite à l'étude des apparitions et disparitions des nœuds au fil du temps).

Dans le cas du réseau étudié ici, le graphe construit à chacune des 2000 itérations est en réalité un arbre (ce qui est lié à la manière dont le réseau est construit, comme expliqué dans la Section 1.1). Cette propriété particulière est mise à profit pour notre visualisation : la source est placée au centre et ses voisins immédiats sont disposés sur un cercle autour d'elle. Les nœuds suivants sont placés sur un cercle de rayon plus important etc. Le rayon du cercle sur lequel se trouve un point indique la distance entre ce nœud et la source (en nombre de sauts).

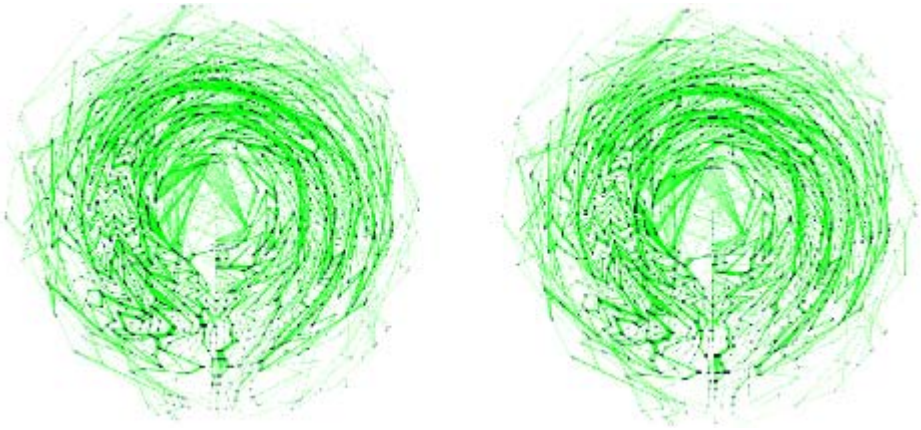

FIG. 2 – *Arbre à 2 itérations successives*

L'étude visuelle de la dynamique du graphe consiste alors en la visualisation (et la comparaison) des graphes correspondant aux cartes successives du réseau. Pour obtenir ces graphes (ici des arbres comme nous l'avons expliqué plus haut), les coordonnées de la totalité des nœuds rencontrés au cours des 2000 itérations ont été fixées[2], et seuls les nœuds et liens présents à une itération donnée sont affichés sur l'arbre correspondant. L'outil *GUESS* a été utilisé pour le dessin du graphe (mais les coordonnées ont été pré-calculées en amont). La Figure 2 illustre l'évolution entre deux itérations ; l'objectif est par exemple de détecter des portions d'arbre qui apparaissent ou disparaissent d'une itération à une autre, traduisant des modifications de routage.

Une première difficulté liée à cette méthode est le choix de la position de chaque nœud, qui a un impact important sur la lisibilité de la visualisation (par exemple à cause des arcs qui se coupent).

Par ailleurs, à moins que les changements de route ne se produisent tout près de la machine source, il est très difficile de distinguer des changements significatifs d'un arbre à un autre, comme le montre la Figure 2. Cette métaphore de visualisation, pourtant naturelle pour représenter une structure arborescente, n'est pas satisfaisante car elle introduit un biais important : les modifications proches de la source sont beaucoup plus facilement observables puisqu'elles modifient une grande portion de l'arbre, à l'inverse des changements concernant les nœuds périphériques.

Cette méthode de visualisation n'est donc envisageable que pour une approche locale ; elle ne pourra être utilisée qu'une fois qu'un sous-ensemble de nœuds et un sous-ensemble d'itérations auront été sélectionnés, c'est-à-dire lorsque l'on saura « où » (quels nœuds) et « quand » (quelles itérations) regarder.

[2] Chaque carte instantanée comprend en moyenne 12000 nœuds. Le nombre total de nœuds rencontrés pendant la période étudiée (2000 itérations) est de l'ordre de 20000.

Approche différentielle

Dans la visualisation étudiée précédemment, chaque carte (arbre) correspond à la vision statique du réseau à un instant donné. Il est possible de refléter un peu de dynamique sur une carte unique en adoptant une représentation que nous qualifions de *différentielle*. Pour une itération donnée, on représente les nœuds qui sont « apparus », c'est-à-dire les nœuds qui étaient absents pendant les *n* itérations précédentes[3], et qui sont présents à l'itération courante. Il est possible d'ajouter une contrainte supplémentaire en imposant que ces nœuds soient présents pendant *m* itérations à partir de l'itération courante (cela permet de distinguer des nœuds apparus de façon très ponctuelle de ceux qui restent présents à partir du moment où ils sont apparus). Réciproquement, on peut s'intéresser aux nœuds qui ont « disparu » à un instant donné (en utilisant également les paramètres *n* et *m*).

Cette représentation différentielle permet de mettre en avant les apparitions et disparitions de nœuds et de liens, et donc de capturer une certaine dynamique sur une représentation unique. Les paramètres *n* et *m* sont respectivement appelés *marge arrière* et *marge avant.*

L'approche différentielle est adaptée pour mettre en évidence une dynamique régulière telle que l'alternance entre deux routes pour l'équilibrage de charge sur le réseau : il suffit pour cela de fixer la marge arrière à 1 (on regarde les nœuds absents à l'itération précédente) et la marge avant à 0 (nœuds présents à l'itération courante) : seuls les nœuds apparus à l'itération courante et absents à l'itération précédente seront affichés. Par contre, les événements « imprévisibles » sont difficilement repérables puisque les valeurs à choisir pour ces deux paramètres ne sont pas connues.

2.2 Matrice d'adjacence

Une première simplification de la métaphore précédente consiste à représenter uniquement les liens (et non plus les liens et les nœuds). La représentation choisie pour cela est la matrice d'adjacence : il s'agit d'une matrice carrée dont les lignes et les colonnes correspondent aux machines rencontrées lors d'au moins une itération sur la période étudiée. Pour une itération donnée, un point est affiché aux coordonnées [i, j] de la matrice s'il existe un lien entre le nœud i et le nœud j à cette itération. Cette représentation est illustrée sur la Figure 3. Tous les nœuds rencontrés au cours des 2000 itérations sont représentés dans la matrice, afin de pouvoir repérer des apparitions ou des disparitions de liens.

Une anomalie de routage peut être supposée lorsqu'une forte modification du nombre de nœuds et/ou de liens est observée entre deux itérations successives (visible par exemple lorsque l'on fait défiler les images les unes après les autres dans une vidéo[4]). Des tels changements se traduisent par l'apparition et la disparition soudaines de points dans la matrice. Ils sont plus aisément détectables si des « blocs » de liens apparaissent ou disparaissent, ce qui n'est visible que si ces liens sont regroupés au sein de la matrice d'adjacence.

[3] *n* est un paramètre à ajuster selon le type d'événement que l'on cherche à mettre en évidence.

[4] Voir la vidéo située à l'url *http://www.complexnetworks.fr/videos.php?video_id=24*.

L'ordre des nœuds dans la matrice est donc important car c'est lui qui rend possible l'apparition de tels blocs. L'objectif est donc de regrouper dans une même zone les machines qui sont proches en terme de topologie de réseau, afin de voir des blocs de liens qui apparaissent ou disparaissent d'une itération à une autre. Le classement des nœuds dans l'ordre croissant de leur adresse IP semble adapté, dans la mesure où les machines d'un même sous-réseau se trouvent ainsi groupées dans la matrice d'adjacence.

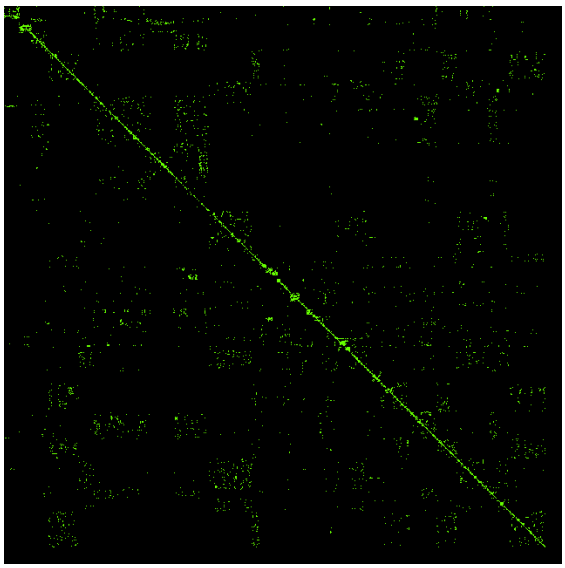

FIG. 3 – *Matrice d'adjacence*

Comme la représentation sous forme de graphe, cette méthode a l'avantage de s'intéresser aux liens entre le machines et pas seulement aux machines elles-mêmes.

Cette représentation permet de faire la distinction entre l'équilibrage de charge, qui se traduit par des clignotements, et les changements brutaux de route, visibles par l'apparition de blocs blancs ou noirs (traduisant les apparitions et les disparitions de liens). Comme nous l'avons expliqué précédemment, ces blocs ne sont visibles que si les liens qui apparaissent ou disparaissent se situent dans une même zone de la matrice, et donc si l'ordre des nœuds est adapté.

Le problème de la taille de la matrice d'adjacence se pose, puisque le nombre total de nœuds distincts présents au cours d'au moins une itération sur les 2000 est d'environ 20000. Afin de pouvoir représenter cette matrice sous une forme plus synthétique, une solution consiste à agréger sur un unique pixel plusieurs valeurs à l'intérieur d'un intervalle d'adresses IP: dans ce cas, un point présent dans la matrice indique qu'au moins un lien existe entre deux machines quelconques de cet intervalle. Cette représentation synthétique implique néanmoins une perte d'information et dépend de la manière dont les coupures sont effectuées entre les différents intervalles d'adresses IP.

Par conséquent, la matrice d'adjacence est elle aussi préconisée pour une représentation locale de la topologie du réseau, pour laquelle on peut se limiter à l'observation d'une portion de la matrice. Comme pour la représentation sous forme de graphe (d'arbre), cela implique en particulier de savoir « où » regarder dans la matrice (i.e. quelles machines).

On peut noter que l'approche différentielle est également possible avec cette métaphore : dans ce cas, un point présent dans la matrice à une itération donnée indique l'apparition d'un lien qui n'existait pas lors des précédentes itérations (ou l'inverse si l'on s'intéresse aux liens qui ont disparu). Les avantages et les inconvénients de cette visualisation différentielle sont les mêmes que pour la représentation d'arbre différentiel proposé dans la Section 2.1.

Bien que très complètes, les visualisations qui représentent les liens entre les machines posent des problèmes en terme d'échelle. Le nombre de liens potentiels entre les 20000 machines observées au cours d'au moins une itération est trop important pour qu'ils puissent être visualisés intégralement, comme nous l'avons vu précédemment. Dans la Section suivante, nous proposons deux métaphores originales, limitées à la représentation des nœuds du réseau.

3 Visualisation des nœuds du réseau

3.1 Courbe de Hilbert

L'objectif de cette représentation est de montrer sous une forme synthétique et affichable sur un écran[5], les machines présentes dans le réseau à une itération donnée. Pour cette visualisation, tous les nœuds rencontrés au cours de la mesure sont ordonnés sur une droite par ordre croissant de leur adresse IP. Afin de passer à une représentation en 2D tout en garantissant que des nœuds proches sur la droite restent proches sur la visualisation, les coordonnées des points sont fixées à l'aide d'une courbe de Hilbert (Figure 4).

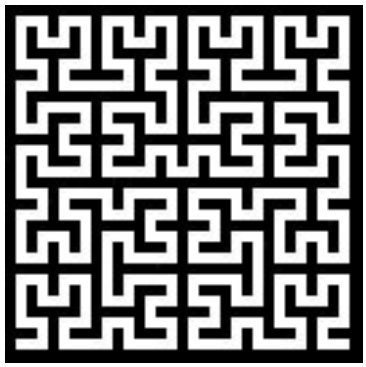

FIG. 4 – *Principe des courbes de Hilbert*

[5] Il s'agit en effet de la principale limite des matrices d'adjacence.

Cette représentation est beaucoup plus synthétique que la matrice d'adjacence, puisque l'on représente uniquement –par des points blancs- les nœuds présents à une itération donnée, au lieu de représenter les liens (Figure 5[6]). Cette métaphore est donc adaptée à une représentation de tous les nœuds. Il s'agit cependant, comme pour les autres visualisations étudiées jusqu'ici, de vues instantanées ; il faut, là encore, savoir « où » regarder (i.e. quelle portion de l'image) et « quand » (quelles itérations). Le repérage d'anomalies dans la dynamique du réseau peut s'effectuer en visionnant les images successivement et en essayant de voir apparaître des blocs d'apparitions ou de disparitions de nœuds. Cependant, l'ordre des nœuds a, là encore, un fort impact sur l'existence de blocs blancs ou noirs et donc sur la détection de modifications du routage.

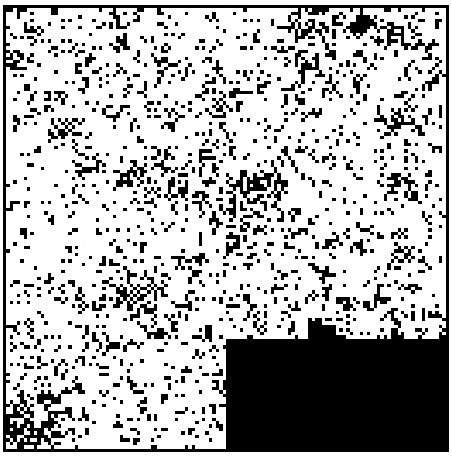

FIG. 5 – *Visualisation « Hilbert »t*

L'approche différentielle est aussi possible avec ce type de représentation : un point sur la courbe de Hilbert est représenté si le nœud correspondant est présent à cette itération alors qu'il était absent aux précédentes (ou inversement).

Les approches différentielles permettent de mettre en évidence la dynamique des nœuds sur une représentation statique (pour une itération donnée). Le paramétrage des marges avant et arrière est cependant difficile. L'idéal serait de représenter la dynamique de toutes les itérations sur une seule représentation statique ; ceci est l'objet de la métaphore présentée dans la Section suivante, dite en « code barre ».

[6] Le nombre total de machines rencontrées pendant la période de mesure n'étant pas une puissance de 2, la Figure 5 comporte une partie « vide » qui correspond au rectangle situé en bas à droite.

3.2 Code barre

La visualisation que nous qualifions de *code barre* (voir les Figures 6 et 7 pour comprendre l'origine de ce nom !) fournit une représentation globale dans le temps, c'est-à-dire qu'elle permet de représenter toutes les itérations sur une même vue.

Sur cette visualisation, la présence d'un point blanc aux coordonnées (i, j) signifie que le nœud j est présent à l'itération i. Comme c'était le cas pour la matrice d'adjacence, cette représentation de 20000 (nœuds) sur 2000 (itérations) est trop volumineuse pour tenir sur un écran ; néanmoins, il est inutile de regarder des images successives pour observer des apparitions ou des disparitions de nœuds (ou idéalement de blocs de nœuds, qui sont plus facilement repérables que des nœuds isolés), ce qui constitue un avantage par rapport à la matrice d'adjacence.

FIG. 6 – *Problème durable*

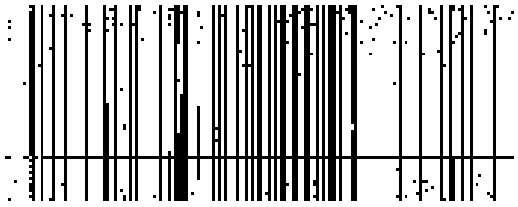

FIG. 7 – *Problème ponctuel*

Cette représentation présente donc un gros avantage sur les métaphores étudiées précédemment : elle capture sur une seule image la dynamique du réseau sur toute la période étudiée. Il s'agit par conséquent de la représentation la mieux adaptée pour identifier les itérations et les machines concernées par des modifications de routage : elle peut permettre de répondre aux questions « où » et « quand » regarder.

Bien que très prometteuse, cette visualisation présente malgré tout quelques faiblesses. Une première limite est liée à sa dépendance à l'ordre des nœuds, comme c'était le cas pour les matrices d'adjacence et pour les courbes de Hilbert.

Une autre faiblesse est liée à la taille importante de l'image générée (de l'ordre de 20000 * 2000), qui ne permet pas de l'afficher dans son intégralité à une résolution suffisante ; il faut par conséquent s'appuyer sur des techniques automatiques pour repérer des blocs dans les données (qui correspondent à un problème relativement durable, comme l'illustre la Figure 6) ou des lignes horizontales (traduisant un problème ponctuel comme le montre la Figure 7).

4 Conclusion et perspectives

Dans cet article, nous avons analysé et comparé plusieurs métaphores de visualisation –classiques ou originales- visant à faciliter la détection d'événements (par exemple d'anomalies) dans des réseaux dynamiques de grande taille. Le réseau particulier étudié ici est un sous-ensemble de l'Internet, constitué de machines et de routeurs, dont l'évolution a été suivie pendant 3 semaines.

L'analyse des forces et faiblesses de ces représentations nous ont conduits à la conclusion suivante quant à l'ébauche d'une méthodologie de détection d'événements dans la dynamique des grands réseaux d'interaction :

- La visualisation en « code barre » est la plus complète du point de vue de la dynamique ; il s'agit donc de la métaphore à utiliser en première intention pour repérer les moments et/ou les machines affectées par des événements inhabituels.

- Les représentations matricielle et sous forme de courbe de Hilbert sont également susceptibles de mettre en évidence des événements, en particulier si un sous-ensemble d'itérations a été identifié par la méthode précédente.

- La visualisation sous forme d'arbre (spécifique au réseau étudié) est la moins adaptée car l'impact d'une modification du routage est extrêmement variable en fonction de l'endroit où elle se produit –en terme de distance à la source.

De nombreuses perspectives s'ouvrent à la visualisation pour la recherche d'événements dans la dynamique des réseaux :

- Il apparaît dans quasiment toutes les métaphores étudiées ici que la notion d'ordre sur les nœuds du réseau a un fort impact sur la mise en évidence de changements dans la dynamique d'un réseau. Un ordre naturel existe pour les adresses IP puisqu'il s'agit de valeurs numériques –même si cet ordre n'est pas forcément significatif lorsque l'on dépasse l'échelle du sous-réseau- mais ça n'est pas le cas pour tous les réseaux (par exemple les réseaux sociaux). Une première perspective consiste donc à généraliser cette notion d'ordre au sein des nœuds d'un réseau d'interaction.

- L'utilisation conjointe de méthodes statistiques et de visualisation représente la deuxième piste que nous souhaitons suivre pour approfondir les travaux présentés dans cet article.

Références

Akamai, Visualizing the Internet, http://www.akamai.com/html/technology/visualizing_akamai.html.

Assia Hamzaoui, Matthieu Latapy and Clémence Magnien, Detecting Events in the Dynamics of Ego-centered Measurements of the Internet Topology, *Proceedings of International Workshop on Dynamic Networks (WDN)*, in conjunction with WiOpt 2010.

V. Batagelj, Visualization of Large Networks, Chapter in *the Encyclopedia of Complexity and System Science* (editor in chief Bob Meyers), in the Complex Networks section (section editor Geoffrey Canright), Springer Verlag, 2009.

K. Claffy, B. Huffaker, Macroscopic Internet visualization and measurement, Caida Mapnet tool, http://www.caida.org/tools/visualization/mapnet/summary.html.

Commetrix, Dynamic Network Visualizsation & Analysis, http://www.commetrix.de/

Welcome To The Blogosphere, Discovery Magazine, May 2007, http://discovermagazine.com/2007/may/map-welcome-to-the-blogosphere

Gephi, the Open Graph Viz Platform, http://gephi.org/

GUESS, the Graph Exploration System, http://graphexploration.cond.org

Clémence Magnien, Frédéric Ouedraogo, Guillaume Valadon, Matthieu Latapy, Fast dynamics in Internet topology: preliminary observations and explanations, *Fourth International Conference on Internet Monitoring and Protection (ICIMP 2009)*, May 24-28, 2009, Venice, Italy.

Soon-Tee Teoh, Kwan-Liu Ma, S. Felix Wu, Dan Massey, Xiaoliang Zhao, Dan Pei, Lan Wang, Lixia Zhang, and Randy Bush, Visual-based Anomaly Detection for BGP Origin AS Change Events, Proc. of the 14th IFIP/IEEE Workshop on Distributed Systems: Operations and Management, 2003.

Soon Tee Teoh, Supranamaya Ranjan, Antonio Nucci, and Chen-Nee Chuah, BGP Eye: A New Visualization Tool for Real-time Detection and Analysis of BGP Anomalies, Proc. of the ACM Conf. on Computer and Communications Security Workshop on Visualization for Computer Security, 2006.

TULIP, Graph visualization Software, http://tulip.labri.fr/

Summary

This paper aims at enabling the visual detection of unexpected « events » in large dynamic networks. Two exhaustive visualization metaphors are studied, which represent the evolution of links over time. Due to scalability issues, we propose two original metaphors restricted to the nodes of the network. The strengths, weaknesses and complementarity of these 4 approaches allow us to propose the first steps of a methodology for the visual detection of anomalies in large dynamic interaction networks.

The methodology and visualizations presented in this article are generic and may be applied to any type of nodes and links; they are illustrated here on a subset of the Internet.

CUBIST Analytics: A Visual Analysis Tool for Formal Concept Analysis

Cassio Melo, Marie-Aude Aufaure

MAS Laboratoire, École Centrale Paris
Grande Voie des Vignes, 92295 Chatenay-Malabry, France
{Cassio.Melo, Marie-Aude.Aufaure}@ecp.fr

Abstract. This paper presents a prototype of *CUBIST analytics*, a Formal Concept Analysis (FCA)-based visual analytics tool for Business Intelligence. The main purpose of *CUBIST analytics* is to provide novel ways of applying visual analytics in which meaningful diagrammatic representations will be used for manipulating, navigating through and visually querying complex data. We present an overview the actual prototype, current challenges and future steps towards an advanced FCA-based visual analytics tool for Business Intelligence.

1 Introduction

The vast amount of data generated by governments and commercial organizations over the last decades has brought new challenges to the information science. There is common agreement that those data may reveal patterns in the real world very valuable to scientific experiments, aerospace industry, business applications and other data-centric applications. However, the exploitation of large amounts of data remains challenging despite the increasing number of analysis methods proposed. In particular, visual analytics aims at bridging this gap by employing more intelligent means in the analysis process and is being used systematically in business applications to help managers and analysts to make better decisions. According to Keim and his colleagues (Keim *et al.* 2008), "visual analytics is an iterative process that involves information gathering, data preprocessing, knowledge representation, interaction and decision making". Visual analytics is therefore, comprised of three main axes: visualization, statistics and data mining.

In order to analyze the relationships between groups of data and their properties, we used Formal Concept Analysis (FCA) as our mainstream analytical framework. Formal Concept Analysis has become popular in early 80's as a mathematical theory of data analysis based on the philosophical notion of a concept (Wille 1984) and since then has been applied to a variety of domains, to name a few, information retrieval (Carpineto & Romano 1995) (Priss 2000); genes expression (Akand *et al.* 2010); machine learning (Kuznetsov 2004). A formal concept is a tuple (E,I), containing an intent (I) and an extent (E) set. The intent set represents all the attributes shared by a set of objects; conversely, the extent set contains all the objects that have the same set of attributes. The analytical framework provided by FCA is traditionally represented by a Hasse diagram with particular properties. The hierarchical structure formed by concepts can provide reasoning for classification, clustering, implication discovery, rule learning (Akand *et al.*). The main problem with the traditional *Hasse* diagram

is the size of the lattice (Priss 2006). When it expands to a few dozens of concepts its comprehension becomes compromised. The problem arises not only for amount of data to be displayed, but it becomes harder and harder to discern between nodes and edges. Solutions to cope with this problem are often referred to as "*clutter reduction*" techniques in the literature (Ellis & Dicks 2007) that is, how to represent the data meaningfully in a limited display.

In this article, we present an overview of our current challenges in lattice visualization, existing work in the domain and particular techniques implemented in our tool *CUBIST Analytics*. We have developed this tool in the context of the CUBIST project ("Combining and Uniting Business Intelligence with Semantic Technologies"), a European project that leverages the state-of-art in Visual Analytics, Formal Concept Analysis and Semantics applied to Business Intelligence in three user cases: space control operations, genes expression analysis and company recruitment process.

2 Background

One of the first visualization tool for FCA lattices was proposed by (Carpineto & Romano 1995) for an information retrieval system called *Galois*. Authors implemented a Fish Eye (Sarkar & Brown 1992) representation of lattices where concept nodes are expanded relatively to their neighbors when users focus on them. Its interface, named *ULYSSES*, allowed users to reduce the search result space by adding constraints to the lattice. Later with *CREDo* only parts of the lattices were displayed similarly to file/folder displays, where a second level of the hierarchy is indented and can be expanded or collapsed interactively by users (Carpineto & Romano 2004). Priss (Priss 2000) used the lattice representation to show the concepts hierarchy in thesauri. Each concept is viewed as a facet in the information retrieval system called *FaIR*. Recently in (Akand *et al.* 2010), authors proposed an algorithm for closed itemsets mining that generates a browsable concept lattice designed for biology applications.

Over the past decades a number of tools have increased systematically. The main purpose of these tools is to generate the concept lattice of a given formal context and the corresponding association rules. *ToscanaJ* (toscanaj.sourceforge.net), *Galicia* (www.iro.umontreal.ca/~galicia) and *Concept Explorer* (*ConExp*) (sourceforge.net/projects/conexp) are among the most popular tools. They can compute and visualize concept lattices but are not designed to do so for large numbers of concepts.

A recent work (Borza et al. 2010) presents *OpenFCA*, an open source FCA-based web application used for context editing, concept computation and visualization, association rules and attribute exploration (code.google.com/p/openfca). Their approach is one of the first tools with highly interactive layout for concept analysis. There are a couple of limitations however, for instance, the only reduction method is based on defining a maximum tree depth and the lattice computation is done on client side. For large lattices *OpenFCA* has a preprocessing tool called *ConfExplore* which performs the lattices computation for large contexts and exports the results to a context file that can be then viewed by the *OpenFCA*. In *CUBIST analytics* all the heavy processing is done on server side transparently, without the intervention of the user for converting data between the client and server. The communica-

tion is based on the Action Message Format - AMF, a robust and one of the most efficient protocol for data exchange (www.jamesward.com/census/). Details of the implementation will be discussed in an upcoming publication.

3 Visual Analytics with *CUBIST Analytics*

The CUBIST project.[1] goal is to provide Business Intelligence (BI) users with visualization and interaction techniques enhanced by semantic technologies and analysis methods such as Formal Concept Analysis (FCA). *CUBIST analytics* is key part of the project, focused on FCA for visual analytics. The visualization of the hierarchical relationship of concepts can be greatly enhanced beyond the standard approaches known from FCA and interlinked with best practices from known BI visualizations. The *CUBIST analytics* tool is currently being developed following a user-centered design paradigm in which target users are actively participating of the whole development process, brainstorming sessions, prototype evaluations and interface refinements. The resulting technology will be demonstrated in three use cases from the domains of market intelligence, computational biology and space control centre operations.

The problem of large lattices visualization is, to some extent, similar to the large graph visualization faced in graph analysis. There is an extensive literature in graph visualization and interaction and well known techniques which can provide a valuable contribution to the lattice visualization (Herman *et al.* 2000). In the following sections we describe the visualization, navigation and interaction techniques employed in *CUBIST analytics* and discuss their advantages and limitations.

Visualization. The first prototype of *CUBIST analytics* has four layouts for displaying the lattice structure: the traditional *Hasse* diagram, the indented tree, the *sunburst* (Stasko 2000) and a Force Directed with a *Hasse* diagram (repulsion forces between edges). It is possible to zoom in and out the layout through the mouse scroll button. The sunburst layout position concept nodes in concentric circles similar to a radial tree with nodes adjusted to shape the concentric space (fig. 1). We used a modified version of the *Prefuse Flare* framework (flare.prefuse.org) for *Flex* to handle the events, structure and animations underneath.

Navigation. The full topology of a lattice seems to be little help in the analytical process (Carpineto & Romano 1995; Priss 2006). The display of the traditional Hasse diagram is only partial for large lattices and we have opted for a combination between keyboard and mouse for a faster and gestural navigation. The user holds the “shift” key to gain an interactive control for scrolling through layout's anchor point in response to the mouse movement (fig. 2). Optionally it is possible to enable the *Fish Eye* (Sarkar & Brown 1992) distortion - it makes the concept node appear bigger when the mouse pointer is over and display its subsequent relatives in a decreasing scale size. This technique is well known in graph navigation literature, known as *focus+context distortion* (Kreuseler & Schumann 1999). It allows the user to focus on items he or she selects without losing the context around (e.g. the concept hierarchy).

[1] “Combining and Uniting Business Intelligence with Semantic Technologies”. Visit www.cubist-project.eu for more information.

Interaction. For interacting with concepts we have implemented a semi-transparent widget menu that pops out when the user clicks on a concept node. This approach is similar to that one used in (McGuffin & Jurissica 2009) and advanced CAD software such as *Maya3D*. This "hot box" is contextual and keeps users focusing on the task without needing to click on a sided toolbar for frequent operations such as "view objects", "view attributes" and to hide parts of the lattice (fig. 3).

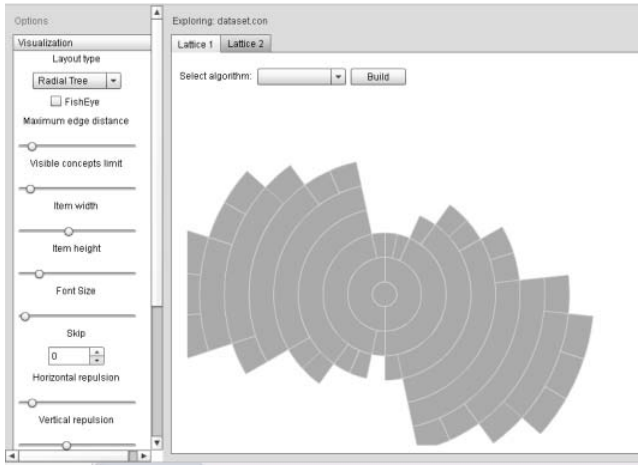

FIG. 1. *Sunburst layout.*

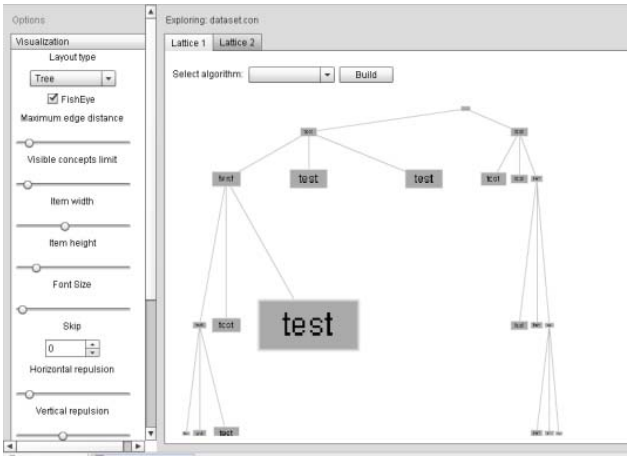

FIG. 2. *Fish eye navigation in the tree.*

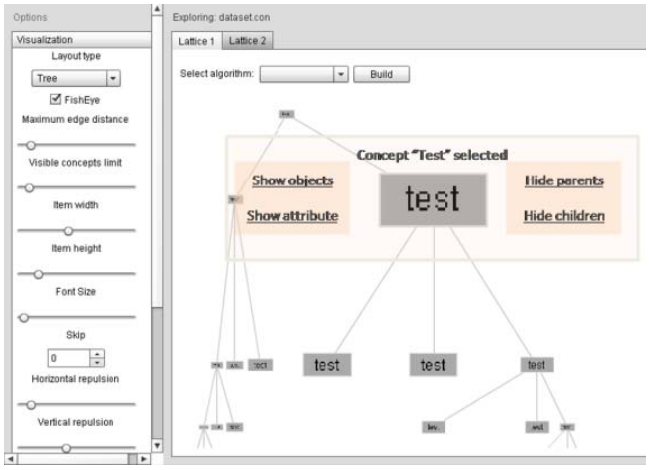

FIG. 3. *A "hot box" is popped out when a concept is clicked*.

4 Open Issues

A couple of questions and remarks appeared during the research presented in this work and it may be fruitful to discuss them.

- We have been discussing whether trees were good candidates for the lattice structure. They are "fully" hierarchical in the sense that a child belongs to only one parent and hence a parent selection criteria would be necessary to transform lattices into trees. We will discuss this issue in another upcoming publication;
- Although it is not the focus of this paper, we have concerns about the performance of drawing algorithms for large lattice exploration. As the visualization is supposed to provide real-time interaction, updates on the lattice structure should be done in a very short time. One solution will probably lead to progressively loading the structure as the user navigates (similarly to *GoogleMaps*), instead of loading it all at once;
- A closer integration with context reduction techniques will be necessary, including techniques such as sub-context mining, noise reduction and clustering;
- Selection and manipulation techniques must also be improved e.g., to allow multiple concepts selection, criteria filter, concept merging and dynamic querying from the lattice.

5 Conclusion

In this work we presented a FCA-based visual analytics tool called *CUBIST analytics*. We believe our approach is a first step to address large lattices visualization. We have implemented a couple of alternatives for lattice exploration; a navigational Fish Eye using combination of a key and the mouse; and an *insitu* "hot box" for concept interaction. Traditional methods for visualization of large lattices are quite restricted to *Hasse* diagrams. New chal-

lenges emerge from the visualization of large lattices and common approaches should be extended to cope with them. The next step consists in exploring visual metaphors such as *pixelization* and tree map as the lattice representation. More sophisticated navigation and interaction techniques for zooming into different levels of concepts granularity are also required. Finally, we hope the future outcomes of this work will help stimulating the FCA community to discuss about lattice visualization issues.

We would like to acknowledge the CUBIST project ("Combining and Uniting Business Intelligence with Semantic Technologies"), funded by the European Commission's 7th Framework Programme of ICT, under topic 4.3: Intelligent Information Management.

Références

Akand E., Bain, M., Temple, M. (2010). *A Visual Analytics Approach to Augmenting Formal Concepts with Relational Background Knowledge in a Biological Domain*, in Sixth Australasian Ontology Workshop (AOW2010).

Borza, P. V. Sabou, O. Sacarea, C. (2010). *OpenFCA: an open source formal concept analysis toolbox*. IEEE International Conference on Automation Quality and Testing Robotics (AQTR 2010).

Carpineto, C., & Romano, G. (1995). *Ulysses: a lattice-based multiple interaction strategy retrieval interface*. In B. Blumenthal, J. Gornostaev & C. Unger (Eds.), Human-Computer Interaction,LNCS 1015-Springer, 91-104.

Carpineto, C., & Romano, G. (2004*). Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO.* Journal of Universal Computing, 10, 8, 985-1013.

Herman, I., Melançon, G., and Marshall, M. (2000). *Graph Visualization and Navigation in Information Visualization: a Survey*.

Keim D. A., Mansmann F., Schneidewind J., Thomas J., and Ziegler H. (2008). *Visual Analytics: Scope and Challenges*. In Visual Data Mining, Lecture Notes In Computer Science, Vol. 4404. Springer-Verlag, Berlin, Heidelberg 76-90.

Kreuseler, M. and Schumann, H. (1999). *Information visualization using a new Focus+Context Technique in combination with dynamic clustering of information space*, Proc. NPIV'99, Missouri, pp. 1-5, 1999, ACM Press.

Kuznetsov, S. (2004). *Machine Learning and Formal Concept Analysis*. In P. Eklund (Ed.), Concept Lattices: Second International Conference on Formal Concept Analysis, LNCS 2961. Berlin: Springer, 287-312.

McGuffin, M. J. & Jurissica, I. (2009). *Interaction Techniques for Selecting and Manipulating Subgraphs in Network Visualizations*. IEEE Transactions on Visualization and Computer Graphics. 15,6, November 2009.

Priss, U. (2000). *Lattice-based Information Retrieval.* Knowledge Organization, 27, 3, 132-142.

Sarkar, M. & Brown, M. H. (1992). *Graphical Fish–eye views of graphs*. In Human Factors in Computing Systems, CHI ’92 Conference Proceedings, ACM Press, pp. 83–91.

Stasko, J. (2000). *An evaluation of space-filling information visualizations for depicting hierarchical structures*. Int. J. Hum.-Comput. Stud. 53, 5 (November 2000), 663-694.

Wolff, K. E. (1984). *A first course in Formal Concept Analysis*, in F. Faulbaum, StatSoft '93, Gustav Fischer Verlag, pp. 429–438

Summary

Cet article présente le prototype de *CUBIST Analytics*, un outil de *Visual Analytics* pour l’Informatique Décisionnelle (Business Intelligence) reposant sur l’Analyse Formelle de Concepts. L’objectif principal de CUBIST est de fournir de nouvelles manières d’appliquer les techniques de *Visual Analytics*, pour lesquelles des représentations appropriées seront utilisées pour la manipulation, la navigation et le requêtage visuel de données complexes. Nous présentons les fonctionnalités du prototype –et les défis résiduels- ainsi que les perspectives de ces travaux.

Gephi, an open source software for visualizing networked data

Sébastien Heymann

LIP6, Université Pierre et Marie Curie - Paris VI
4 place Jussieu, 75252 Paris, France
Sebastien.heymann@gmail.com

Abstract. Gephi is an open-source network visualization platform. It aims to create a sustainable software and a technical ecosystem driven by an international open-source community, who shares common interests in networks and complex systems. Designed to make data navigation and manipulation easy, it aims to fulfill the complete chain from data importing to aesthetics refinements and interaction.
Users interact with the visualization and manipulate structures, shapes and colors to reveal hidden properties. The goal is to help data analysts to make hypothesis, intuitively discover patterns or errors in large data collections.
The rendering engine can handle networks larger than 100K elements and guarantees responsiveness. In addition of interactive exploration, Gephi embed most critical metrics used in Social Network Analysis, including Betweenness, Clustering Coefficient, PageRank or Modularity. More metrics can be added thanks to the extensible software architecture and the open-source code. Focus is also made on interoperability, as Gephi can open major file formats, including GraphML, UCINet DL or Pajek. Network results can be exported as PNG, SVG and PDF. Most of current development efforts are made on Dynamic Network Analysis (DNA). Gephi already provides a Timeline component to study network evolutions and visualize changes over time.
Created with the idea to be the Photoshop of network visualization, our approach is to provide a visual tool with a smooth learning curve and an active open-source community supporting the project. Many efforts have been made to facilitate the community growth, by providing tutorials, plug-ins development documentation, support and student projects. The modular architecture allows any researcher or developer to extend, reuse and mashup Gephi features in different forms.

Gephi, an open source software for visualizing networked data