14e édition des Journées francophones

EGC 2014

Extraction et Gestion des Connaissances

28-31 Janvier

RENNES

IRISA & Centre Inria Rennes - Bretagne Atlantique

Campus de Beaulieu, Rennes

Journées Ateliers/Tutoriels

FGG : Fouille de Grands Graphes

Atelier Fouille de Grands Graphes

Organisateurs : Lydia Boudjeloud-Assala (LITA EA 3097, Université de Lorraine), Bénédicte Le Grand (CRI EA 1445, Universite Paris 1 Panthéon - Sorbonne), Rushed Kanawati (LIPN CNRS, Université Paris 13)

PRÉFACE

Le groupe de travail Fouille de Grands Graphes a été créé en 2010, il s'intéresse à l'analyse et de l'étude de la dynamique dans les grands graphes. L'objectif du groupe est de proposer une structure d'animation scientifique pour des chercheurs venant de plusieurs disciplines et s'intéressant à la fouille de grands graphes. La recherche en modélisation, en analyse et en fouille de grands graphes a connu un net regain d'intérêt qui peut se justifier par les deux points suivants : dans plusieurs domaines les données se présentent souvent sous forme structurée : graphes ou objets reliés. Nous pouvons citer par exemple les systèmes biologiques, le web, les réseaux sociaux (facebook, twitter, ...) ou encore les réseaux bibliographiques. Les graphes issus de ces domaines ont des propriétés spécifiques qui les différencient des graphes aléatoires (graphes sans échelle, faible densité, faible degré de séparation, ...). Les techniques actuelles permettent d'observer de très grands réseaux qui évoluent dans le temps et nous posent ainsi le défi du passage à l'échelle et la nécessité de disposer d'outils de visualisation et de fouille pouvant s'adapter à ce nouveau type de données.

Suite au succès des 4 journées thématiques Fouille de Grands Graphes du groupe de travail à Toulouse'2010, Grenoble'2011, Villetaneuse'2012 et Saint-Etiennes'2013, le groupe de travail EGC Fouille de Grands Graphes (EGC-FGG) propose un atelier AFGG-EGC'2014 pour sa deuxième édition, dans le cadre de la conférence Extraction et Gestion de Connaissances (EGC'2014). En 2012, le groupe de travail FGG et le groupe de travail AFIHM-EGC Visualisation d'information, interaction et fouille de données (GT-VIF) ont proposé un atelier commun. Cet atelier a pour but de réunir les chercheurs intéressés par le traitement, la fouille et la visualisation de grands graphes. Notre ambition est de permettre aux participants d'aborder tous les thèmes de la fouille de grands graphes. L'atelier concerne aussi bien les chercheurs du monde académique que ceux du secteur industriel, et autant les notions conceptuelles que les applications. Le thème principal pour cette édition est lié aux communautés egocentrées, abordées notamment dans une présentation invitée. Cet atelier s'intéressera aussi aux réseaux sociaux et complexes, une application autour des réseaux bibliographiques sera présentée.

Nous tenons à remercier tout d'abord le pôle ResCom et le GDR ASR CNRS pour leur soutien financier. Nous remercions également les auteurs, les membres du comité de programme pour la qualité de leurs contributions ainsi que les responsables des ateliers d'EGC 2014. Enfin nous remercions vivement les présidents : Chantal Reynaud, présidente du comité de programme, Arnaud Martin et René Quiniou coprésidents du comité d'organisation d'EGC 2014.

Lydia Boudjeloud-Assala	Bénédicte Le Grand	Rushed Kanawati
Université de Lorraine,	Université Paris 1, Panthéon-Sorbonne,	Institut Galilée, Université Paris Nord,
LITA-EA 3097	CRI-EA 1445	LIPN CNRS UMR 7030

Membres du comité de lecture

Le Comité de Lecture est constitué de:

Hanane Azzag, Université Paris 13, LIPN
Lydia Boudjeloud-Assala, Université de Lorraine, LITA EA 3097
Eric Fleury, ENS Lyon / INRIA
Alain Gely, Université de Lorraine, LITA EA 3097
Jean-Loup Guillaume, LIP6 - UPMC
Rushed Kanawati, Université Paris 13 - LIPN
Bénédicte Le Grand, Université Paris 1 - CRI
Mustapha Lebbah, Université Paris 13, LIPN-CNRS
Clémence Magnien, LIP6 (CNRS - UPMC)
Maria Malek, LARIS EISTI - Campus de Cergy
Fabien Picarougne, LINA, University of Nantes
Bruno Pinaud, CNRS UMR 5800 LaBRI, INRIA Bordeaux Sud-Ouest
Christophe Prieur, LIAFA, Paris, France
Guillaume Santini, Université Paris 13
Fabien Tarissan, Université Pierre et Marie Curie (UPMC), Paris, France
Emmanuel Viennet, L2TI, Institut Galilée, Université Paris 13
Nathalie Villa-Vialaneix, Institut de Mathématiques de Toulouse

TABLE DES MATIÈRES

Communautés ego-centrées

Jean-Loup Guillaume*

*Sorbonne Universités, UPMC Univ Paris 06 - CNRS, UMR 7606, LIP6
jean-loup.guillaume@lip6.fr

Résumé

L'explosion de la taille des réseaux complexes tels que Facebook ou Wikipedia offre de nouvelles opportunités d'établir de nouveaux contacts et de partager de l'information et des savoir. L'excès d'information peut cependant être également néfaste : sur Facebook, à quels contacts envoyer un message sachant que la plupart des utilisateurs ont plusieurs centaines d'amis ? Quelles pages Wikipedia faut-il lire en priorité pour apprendre le plus possible sur un sujet donné. Ces deux exemples illustrent deux problèmes classiques en analyse de réseaux complexes et en fouille de données : la détection de communautés et le problème de classement.

Les communautés sont des groupes de sommets fortement connectés et les approches classiques peuvent se classer en plusieurs groupes :

- Les méthodes de partitionnement pour lesquelles les sommets appartiennent à une unique communauté. Cette vision est facilement compréhensible, mais reste limitée car, dans la plupart des contextes, les sommets appartiennent à plusieurs communautés.
- Les méthodes recouvrantes qui sont beaucoup plus proches de la réalité, mais qui posent en pratique de nombreux problèmes de définition et de calcul.
- Les méthodes ego-centrées qui se limitent à la recherche des communautés d'un sommet et sont donc un compromis plus accessible à l'heure actuelle.

Je vais présenter dans cet exposé des réflexions sur cette troisième vision. Dans un premier temps, et afin de contourner les problèmes classiques des fonctions de qualité, je présenterai deux fonctions de proximité, une basée sur la dynamique d'opinion et la seconde basée sur la proximité topologique. Toutes deux permettent d'identifier rapidement les sommets proches d'un sommet d'intérêt et ont chacune un intérêt spécifique.

Je présenterai ensuite le concept de communauté multi-ego-centrée, c'est-à-dire de communauté centrée sur un ensemble de noeuds. Je montrerai qu'il est possible avec les fonctions de proximité, notamment celles introduites précédemment, de répondre à deux problèmes proches : (i) comment identifier la communauté contenant un ensemble de sommets fixés et (ii) comment extraire toutes les communautés autour d'un sommet. Je présenterai enfin quelques éléments de validation sur un jeu de données constitué de pages Wikipedia.

Biographie

Jean-Loup Guillaume est maître de conférences HDR à l'Université Pierre et Marie Curie (Paris 6) depuis 2007. Il a obtenu sa thèse de doctorat de l'Université Paris Diderot (Paris 7) en 2004 à la suite de quoi il a effectué des post-doctorats à France Télécom et à l'Université Catholique de Louvain. En 2007, il a rejoint le LIP6 pour participer au développement d'une équipe de recherche sur les réseaux complexes. Ses intérêts en recherche sont centrés autour des réseaux complexes, notamment les aspects liés à leur dynamique. Ses deux thèmes de recherche actuels sont l'étude des phénomènes de diffusion dans les réseaux complexes et leur structure communautaire (dynamique). Il est notamment co-auteur de la "méthode de Louvain" qui est à l'heure actuelle la méthode la plus rapide et la plus utilisée dans le monde pour le calcul de communautés.

Combinaison de modularités locales pour l'identification de communautés égo-centrées

Rushed Kanawati

LIPN - UMR CNRS 7030,
Institut Galilée, Université Paris Nord
99 Av. J-B. Clément, 93430 Villetaneuse
prénom.nom@lipn.univ-paris13.fr
http://www-lipn.univ-paris13.fr/A3

Résumé. Une approche classique pour l'identification de communauté égo-centrée d'un nœud dans un graphe de terrain consiste à appliquer un algorithme d'optimisation gloutonne d'une fonction de qualité donnée. Différentes fonctions de qualité, dites aussi modularités locales, ont été proposées dans la littérature. Nous proposons ici une simple approche de combinaison des modularités locales en utilisant des algorithmes classiques d'*ensemble ranking*. Nous expérimentons notre approche sur des réseaux réels de benchmark. Les premiers résultats obtenus sont encourageants et montrent la validité de l'approche.

1 Introduction

Complex networks are frequently used for modeling interactions in real-world systems in diverse areas, such as sociology, biology, information spreading and exchanging, scientometrics and many other different areas. One key topological feature of real-world complex networks is that nodes are arranged in tightly knit groups that are loosely connected one to each other. Such groups are called *communities*. Nodes composing a community are generally admitted to share common proprieties and/or be involved in a same function and/or having a same role. Hence, unfolding the community structure of a network could give us much insights about the overall structure of a complex network. A major trend in this area was devoted to computing disjoint communities where one node can only belong to one community. A large amount of different algorithms for graph partitioning has been proposed. Some comprehensive survey studies are provided in (Fortunato, 2010; Tang et Liu, 2010). However, in real-world settings, nodes may belong to different communities at once. Some overlapping community detection algorithms have been proposed (Palla et al., 2005; Whang et al., 2013; Shi et al., 2013). A recent survey on overlapping community detection problem is provided in (Xie et al., 2013). Both problems, disjoint and overlapping communities detection are NP-hard (Brandes et al., 2008). Existing algorithms apply different heuristics. The large scale of today available graphs makes most of existing approaches very time consuming. However, in a number of concrete situations, we seek to detect the community of a target node in the network rather than decomposing the whole network into communities. This is for instance, the case of recommender systems where the goal is to identify the set of most similar nodes to one given

node. We denote such a community, computed around a given node, *ego-centered community*. The term of *local community* is also used to refer to the same concept (Clauset, 2005).

A main trend in the area of ego-centered community identification consists on applying greedy optimization approaches guided by a given quality function. Starting from the query node, the local neighborhood is explored. The best node that maximise the applied quality function is *selected* (or elected) to be added to the community structure. Different quality functions has been proposed (Clauset, 2005; Bagrow et Bollt, 2005; Chen et al., 2009). The process of neighborhood exploration ends when there is no more nodes that can enhance the applied quality criteria.

In this work we explore applying ensemble ranking approaches in order to combine different quality functions in the process of detecting ego-centered communities. The second approach consists on using a bench of quality functions at once each providing a rank for the list of candidate nodes to be added to the ego-centered community of the query node. Ensemble ranking approaches are applied to merge the different obtained rankings (Dwork et al., 2001; Chevaleyre et al., 2007; Wei et al., 2010). This requires to change the stopping criteria of a classical greedy optimisation algorithm since different quality functions yields non comparable quality values.

The reminder of this paper is organized as follows. Next in section 2, we introduce basic notations used in this paper. A quick survey of state of the art of ego-centered community detection algorithms is given in 3. Ensemble ranking based approach is detailed in section 4. Experiments on benchmark networks are described and commented in section 5. Finally we conclude and provide main perspective of this work in section 6.

2 Problem definition & notations

Let $G =< V, E >$ be an undirected simple graph defined over a set of nodes V. E is the set of edges in G. Let $v_q \in V$ be the query node for which an ego-centered community should be computed. We denote by $\Gamma(v)$ the set of direct neighbors of a node $v \in V$. We define the extended neighborhood of a node $v \in V$ as $\widehat{\Gamma}(x) = \Gamma(x) \cup \{x\}$. Let $X \subseteq V$ be a subset of V, the complement of X is denoted by $\overline{X}$.

A greedy optimization algorithm starts to explore the network from the query node v_q. Let D be the set of current explored nodes. We can classify nodes in V at any time during the exploration process, into three disjoint sets :

- *The core set (denoted by C)* : is composed of explored nodes whose all neighbors are also explored. In a formal way. We have $C = \{x \in V s.t. \widehat{\Gamma}(x) \subseteq D\}$.
- *The border set (denoted B)* : is composed of explored nodes that have at least one unexplored neighbor node. Formally, $B = \{x \in D \ : \ \exists v \in \Gamma(x) : v \notin D\}$.
- *The shell set (denoted by S)* : is composed of nodes partially explored. These are nodes that have some neighbors in the set B. Formally, $S = \{x \in \overline{D} : \Gamma(x) \cap D \neq \phi\}$.
- *The set of unexplored nodes (denoted by U)* : This is the set of nodes in V that are not explored at all. Formally, $U = \{x \in \overline{D \cup S}\}$.

Notice that D is equal to $B \cup C$. Figure 1. illustrates the different sets of nodes at a given time t during the exploration process. The goal of an ego-centered community detection algorithm is to compute the most relevant set $C \cup B$. Different evaluation criteria can be ap-

plied to measure the relevancy of a computed community. Most used evaluation approaches are summarized in section 5.

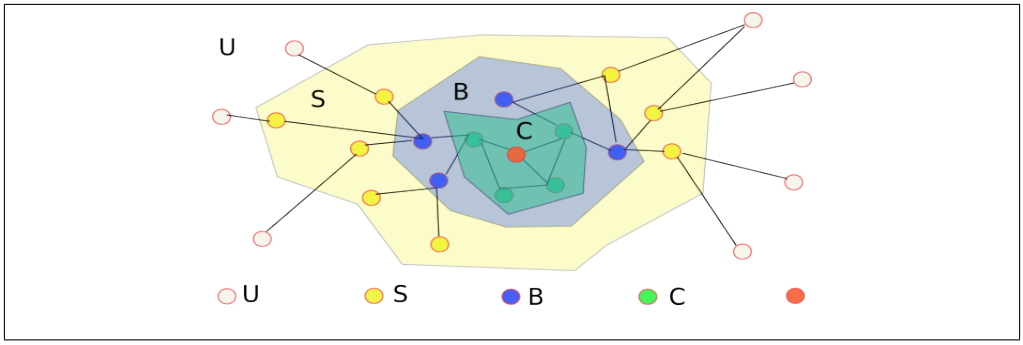


FIG. 1: Illustration of the definitions of the different sets of nodes during the exploration the neighborhood of a query node v_c

3 Ego-centered community identification approaches

Greedy optimisation constitue the main approach for local communities identification. This is implemented as an iterative process. Initially, C is set to the singleton $\{v_c\}$, B is initialized to the empty set and S is set to $\Gamma(v_c)$. Then, at each iteration, nodes in S are ranked according to a quality function Q. The top ranked node is added to the set B. Then all three sets C, B and S and the algorithm iterates while S is not empty and while the quality induced by the selected node increase. Algorithm 1 gives the general outlines of a basic greedy optimization local community detection approach.

In (Clauset, 2005), authors proposed a quality function that focus on the separability of the computed community from other components of the graph. This function, denoted the R modularity is given by :

$$R = \frac{B_{in}}{B_{in} + B_{out}} \tag{1}$$

where B_{in} is the number of links between vertices in B and other vertices in either B and C. B_{out} is the number of links between vertices in B and those in S. A similar local modularity, called the M modularity, is proposed in (Luo et al., 2008). This is given by the following formula :

$$M = \frac{D_{in}}{D_{out}} \tag{2}$$

where D_{in} is the number of ties linking nodes in either B or C, and D_{out} is the number of out-going links from B to S.

Another quality metric guided by the internal density of the computed community is proposed in (Chen et al., 2009). This quality function denoted by L is given by the ratio of the average internal degree of nodes inside the community ($\sum_{i \in C \cup B} |\Gamma(i) \cap (B \cup C)| \times (|C \cup B|)^{-1}$) to the average out degree to nodes in the border set B. ($\sum_{j \in B} |\Gamma(j) \cap S| \times |B|^{-1}$). Formally this is given by the following formula :

Algorithm 1 Greedy optimisation ego-centered community identification algorithm

Require: $G =< V, E >$ a connected graph, $v_c \in V$ a query node, Q a quality function
{/* initializations */}
$C \leftarrow \{v_c\}$
$B \leftarrow \emptyset$
$S \leftarrow \Gamma(v_c)$
$Q \leftarrow 0$
repeat
  $v_s \leftarrow \text{argmax}_{v \in S} Q(v)$
  $\hat{Q} \leftarrow Q(v_s)$
  if $\hat{Q} >= Q$ **then**
    $B \leftarrow B \cup \{v_s\}$
    $S \leftarrow \Gamma(v_s)/(B \cup C)$
    for all $b \in B$ **do**
      if $\Gamma(b) \cap S == \emptyset$ **then**
        $C \leftarrow \{b\}$
        $B \leftarrow B - \{b\}$
      end if
    end for
    $Q \leftarrow \hat{Q}$
  end if
until $S == \emptyset$ **or** $\hat{Q} < Q$
return $B \cup C$

$$L = \frac{\sum_{i \in C \cup B} |\Gamma(i) \cap (B \cup C)| \times (|C \cup B|)^{-1}}{\sum_{j \in B} |\Gamma(j) \cap S| \times |B|^{-1}} \tag{3}$$

Other similar quality functions has also been proposed in (Bagrow, 2008; Ngonmang et al., 2012; Wu et al., 2012). Ameliorations of the basic greedy optimizations has been also proposed introducing a phase of some nodes removal as proposed in (Wu et al., 2012). Performances of greedy optimisation algorithms depends heavily on the position of the query node in the network. Whether the query node is in the core of a given community or at the crossing of different ones. One alternative to overcome this problem consists on searching a seed node that is close to the query node. The community is then expanded around the selected seed node (Lim et Datta, 2013; Ma et al., 2013; Zhang et Wu, 2012).

In (Danisch et al., 2013), authors propose a completely different approach for local community detection based on measuring vertex similarity in the network. The key idea of the algorithm is based on the assumption that if a group of nodes are equally similar to a query node v_c, this groupe constitues the community of v_c. Since a node can blening to different communities at once, authors propose to measure the similarity of all nodes in the graph with respect to the query node but also to a similar node selected to be similar enough to v_c. While the idea is appealing, this algorithm requires computing similarities between all nodes of the graph which is prohibitif in large-scale networks.

4 The proposed approach

4.1 General description

The basic idea here is to apply the same greedy optimisation algorithm as described in algorithm 1, but after replacing the node selection criteria to be moved from the S set to the B set (line 6 in algorithm 1). We propose using k different local modularity functions. Each evaluating the gain from adding a node to the B set form a different point of view. We search to select the node from S that enhance as much as possible all the selected quality functions. This is done as follows :

Each modularity function Q_i ranks elements in the shell set S differently. Let S^{Q_i} be the ranked list of elements of S in function of Q_i. The winner node to be moved from S to B is taken to be the winner after merging the different ranked lists $\{S^{Q_1}, \ldots, s^{Q_k}\}$. Different functions for rank aggregation (also known as voting algorithms) defined in the context of the computational social theory (Dwork et al., 2001; Chevaleyre et al., 2007) can be applied. Let $v_w \in S$ be the winner node : the node that is ranked first after the rank merging process. Let $Q_i(v_w)$ be the i^{th} modularity obtained from adding v_w to B. Different stopping policies can be applied :

- *Strict policy* : where all the k modularities related to the winner node v_w are greater or equal to the respective modularities computed in the previous iteration. This strategy will led to identifying very small communities around the query node.
- *Majority policy* : where the expansion process continues while more than half of used modularities give better or equal qualities for the winner node than for previously elected winner (in the previous iteration). This is a kind of a relaxation of the first strategy and it allows detecting bigger communities than those detected using the first strategy.
- *Least gain policy* : The algorithm iterates if there exist at least one modularity that is enhanced or equal to the same modularity computed for the previously winner node selected in the previous iteration. This is the weakest strategy we can apply allowing to detect large size communities.

Given two lists of modularities values QV_i and QV_j. We denote by $QV_i \succ_p QV_j$ the fact that list QV_i is preferred to QV_j according to policy p. The proposed algorithm is detailed in 2.

4.2 Ensemble Ranking

Ensemble ranking (a.k.a rank aggregation, rank fusion) is the problem of combining a number of ranked lists with same elements in order to get a single list with all elements in it (Dwork et al., 2001). Rank aggregation methods can be *position-based* or *order-based*. Score-based methods use score information from individual rankers while order-based methods use only the rank information.

One well known position-based method is Borda's method Borda (1781) : A Borda score is computed for each element in the lists. For a set of complete ranked lists $L = [L_1, L_2, L_3, ..., L_k]$, the Borda's score of an element i and a list L_k is given by : $B_{L_k}(i) = \{count(j) | L_k(j) < L_k(i) \& j \in L_k\}$. The total Borda's score for an element is then : $B(i) = \sum_{t=1}^{k} B_{L_t}(i)$. Elements are sorted in function of their total Borda score with random selection in case of ties.

Algorithm 2 Combined Modularities for ego-centered community detection algorithm

Require: $G =< V, E >$ a connected graph,
$v_c \in V$ a query node,
$\mathcal{Q} = \{Q_1, \ldots, Q_k\}$ a set of k quality functions
a stopping policy p {/* initializations */}

$C \leftarrow \{v_c\}$
$B \leftarrow \emptyset$
$S \leftarrow \Gamma(v_c)$
for $i = 0$ **to** k **do**
  $QV[i] \leftarrow 0$
end for
repeat
  for all $Q_i \in \mathcal{Q}$ **do**
    compute S^{Q_i}
  end for
  $S^* \leftarrow$ **Rank-aggregation**$(S^{Q_1}, \ldots, S^{Q_k})$
  $v_s \leftarrow S^*[0]$
  for $i = 0$ **to** k **do**
    $\widehat{QV}[i] \leftarrow Q_i(v_s)$
  end for
  if $\widehat{QV} \succ_p QV$ **then**
    $B \leftarrow B \cup \{v_s\}$
    $S \leftarrow \Gamma(v_s)/(B \cup C)$
    for all $b \in B$ **do**
      if $\Gamma(b) \cap S == \emptyset$ **then**
        $C \leftarrow \{b\}$
        $B \leftarrow B - \{b\}$
      end if
    end for
    $QV \leftarrow \widehat{QV}$
  end if
until $S == \emptyset$ **or** $\widehat{QV} \prec_p QV$
return $B \cup C$

A Kemeny optimal aggregation (Kemeny, 1959) is an aggregation that has the minimum number of pairwise disagreements with all rankers. Pairwise disagreement is computed by the *Kendall tau* distance (Lapata, 2006). Computing an optimal Kemeny aggregation is NP-hard starting from a list of four candidates. Different approximate Kemeny aggregation approaches have been proposed in the literature. The basic idea of all proposed approximate Kemeny aggregation is to sort the candidate list, using standard sorting algorithms, but using a non transitive comparison relationship between candidates. This relation is the following : s_i is preferred to s_j, noted $s_i \succ s_j$, if the majority of rankers ranks s_i before s_j. Since the $\succ$ relation is not transitive, different sorting algorithms will provide different rank aggregations with different proprieties. In (Dwork et al., 2001) authors propose a *local Kemeny* aggregation applying a bubble sort algorithm. In (Melville et al., 2010) authors propose an *approximate Kemeny* aggregation applying quick sort algorithm. Further discussion of ensemble ranking approaches is out of the scope of this work.

5 Experiments

In order to quantitatively analyse and compare performances of the different proposed approaches we have applied these to networks whose community structure is already known. We mainly follow the evaluation procedure proposed in (Bagrow, 2008). Let $P_{real} = \{C_1, \ldots, C_m\}$ the real partition of a network vertices set into m communities. Let C_v^Q be the local community of node $v \in V$ computed applying a given method Q. The set of vertices V can then be partitioned into two sets :

$$P_Q^v = \{C_v^Q, \overline{C_v^Q}\}$$

where $\overline{C_v^Q}$ denotes the complement of set C_v^Q. On another hand, for each $v \in V$ we derive a bi-set real partition from P_{real} as follows :

$$P_{real}^v = \{P_j : v \in P_j, \bigcup_{i \neq j} P_i\}$$

The quality of the computed community C_v^Q is then measured by the similarity between both clusterings P_{real}^v and P_Q^v. Let sim be a clustering similarity measure. The overall performance of a local community identification method Q can then be simply given by :

$$\frac{\sum_{v \in V} sim(P_Q^v, P_{real}^v)}{|V|}$$

Different clusterings comparaison or similarities functions have been proposed in the literature (Aggarwal et Reddy, 2014). Next we apply two widely used indices : the Adjusted Rand Index (ARI) (Hubert et Arabie, 1985) and the Normalized Information Index (NMI) (Strehl et Ghosh, 2003).

The ARI index is based on counting the number of pairs of elements that are clustered in the same clusters in both compared partitions. Let $P_i = \{P_i^1, \ldots, P_i^l\}$, $P_j = \{P_j^1, \ldots, P_j^k\}$ be two partitions of a set of nodes V. The set of all (unordered) pairs of nodes of V can be partitioned into the following four disjoint sets :

- S_{11} = { pairs that are in the same cluster under P_i and P_j }
- S_{00} = { pairs that are in different clusters under P_i and P_j }
- S_{10} = { pairs that are in the same cluster under P_i but in different ones under P_j }
- S_{01} = { pairs that are in different clusters under P_i but in the same under P_j }

Let $n_{ab} = |S_{ab}|, a, b \in \{0, 1\}$, be the respective sizes of the above defined sets. The rand index, initially defined in (Rand, 1971) is simply given by :

$$\mathcal{R}(P_i, P_j) = \frac{2 \times (n_{11} + n_{00})}{n \times (n-1)}$$

In (Hubert et Arabie, 1985), authors show that the expected value of the Rand Index of two random partitions does not take a constant value (e.g. zero). They proposed an adjusted version which assumes a generalized hypergeometric distribution as null hypothesis : the two clusterings are drawn randomly with a fixed number of clusters and a fixed number of elements in each cluster (the number of clusters in the two clusterings need not be the same). Then the adjusted Rand Index is the normalized difference of the Rand Index and its expected value under the null hypothesis. It is defined as follows :

$$ARI(P_i, P_j) = \frac{\sum_{x=1}^{l} \sum_{y=1}^{k} \binom{|P_i^x \cap P_j^y|}{2} - t_3}{\frac{1}{2}(t_1 + t_2) - t_3} \tag{4}$$

where :

$$t_1 = \sum_{x=1}^{l} \binom{|P_i^x|}{2}, t_2 = \sum_{y=1}^{k} \binom{|P_j^y|}{2}, t_3 = \frac{2t_1 t_2}{n(n-1)}$$

This index has expected value zero for independent clusterings and maximum value 1 for identical clusterings.

Another family of partitions comparisons functions is the one based on the notion of mutual information. A partition P is assimilated to a random variable. We seek to quantify how much we reduce the uncertainty of the clustering of a randomly picked element from V in a partition P_j if we know P_i. The Shanon's entropy of a partition P_i is given by :

$$H(P_i) = -\sum_{x=1}^{l} \frac{|P_i^x|}{n} log_2(\frac{|P_i^x|}{n})$$

Notice that $\frac{|P_i^x|}{n}$ is the probability that a randomly picked element from V be clustered in P_i^x. The mutual information between two random variables X,Y is given by the general formula :

$$MI(X, Y) = H(X) + H(Y) - H(X, Y) \tag{5}$$

This can then be applied to measure the mutual information between two partitions P_I, P_j. The mutual information defines a metric on the space of all clusterings and is bounded by the entropies of involved partitions. In (Strehl et Ghosh, 2003), authors propose a normalized version given by :

$$NMI(X, Y) = \frac{MI(X, Y)}{\sqrt{H(X)H(Y)}} \tag{6}$$

Another normalized version is also proposed in (Fred et Jain, 2003). Other similar information-based indices are also proposed (Nguyen et al., 2009; Meila, 2003).

In a first expriment, we have compared the average NMI and ARI indices obtained from applying simple local modularities R, M, L and the different proposed ensemble ranking versions : Borda, Approximate Kemeny and Local Kemeny. For ensemble ranking approaches we tested the outputs for the three different stoping policies : strict, majority (denoted maj) and least gain (denoted LG) (see section 4.1). All variations are applied on a bench of classical benchmark networks. In table 1 we summarize basic characteristics of selected benchmark real networks.

TAB. 1: Characteristics of some well-known benchmark networks

Network	n	m	**# com**	**reference**
Zachary club	34	78	2	(Zachary., 1977)
Football	115	616	11	(Girvan et Newman, 2002)
Political books	100	441	3	(Krebs)
Dolphins	62	159	2	(Lusseau et al., 2003)

Due to the randomness in node selection phase in case of ties, we have executed each method 10 times and we show on next figures the average NMI and the average ARI values along with the standard deviation values.

For all networks, at least one ensemble clustering approach give better results than applying simple local modularities. For almost all networks the Kemeny-based approaches give better results than Borda, except for the american football network. Surprisingly, for this later network, the Borda method gives a high value for both NMI and ARI when applying the least gain policy (LG). More investigations should be conducted to explain this results on this particular case. In general, obtained NMI and ARI values are rather small. This is mainly due to the averaging scheme since for many nodes local communities may differ in significant way from the global known global community. For all tested networks, the least gain stopping policy yield the best results compared to the two other policies. This policy allows detecting rather large communities that approximate better than the others policies global communities given by the ground-truth partition. The standard deviation of all ensemble ranking approaches is nearly null for all datasets. This is not the case of basic approaches. Notice that optimizing the M local modularity yields good average but with much higher standard deviation. This shows that ensemble ranking approches provide more stable results and hinder the effect of random election of nodes in case of ties during the greedy optimisation process.

These first obtained results are encouraging and show the potential of applying ensemble ranking to enhance the detection of ego-centered communities. We are working actually on confirming these results on artificial networks generated using the LFR network generator that allow to fine-tune major characteristics of test networks (Lancichinetti et Radicchi, 2008).

The results need also to be examined in a more detailed way in function of the position of each node in the network. It would be interesting to mine an association between node's centrality and the method giving the best local community. This may allow to apply weighted

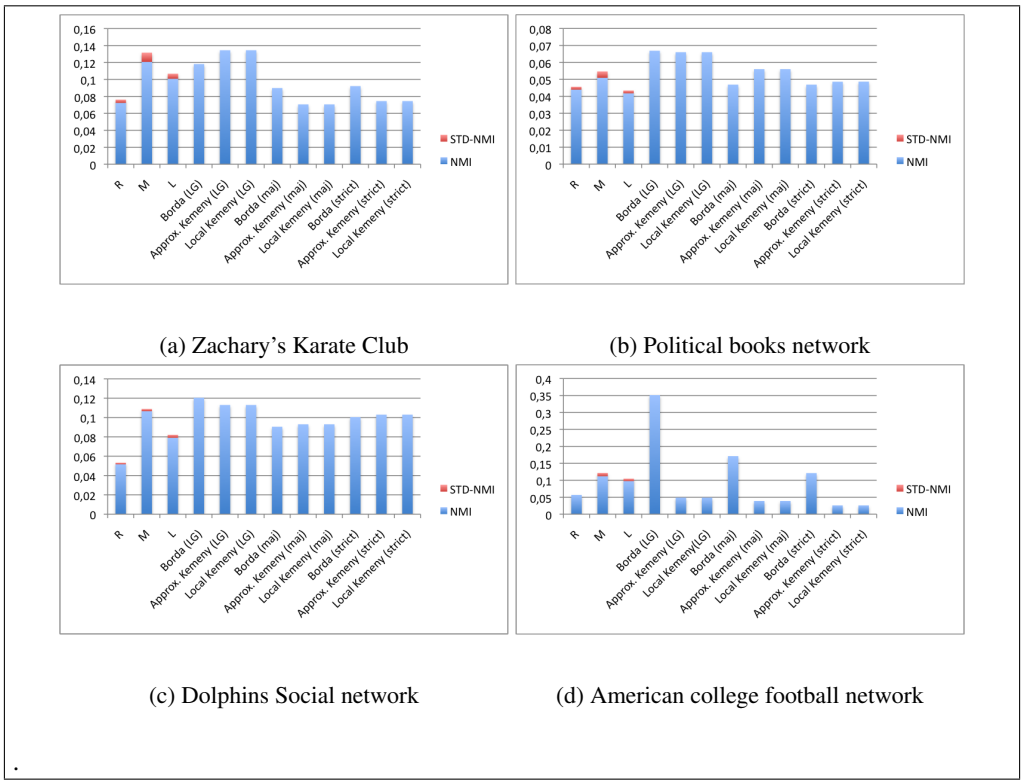
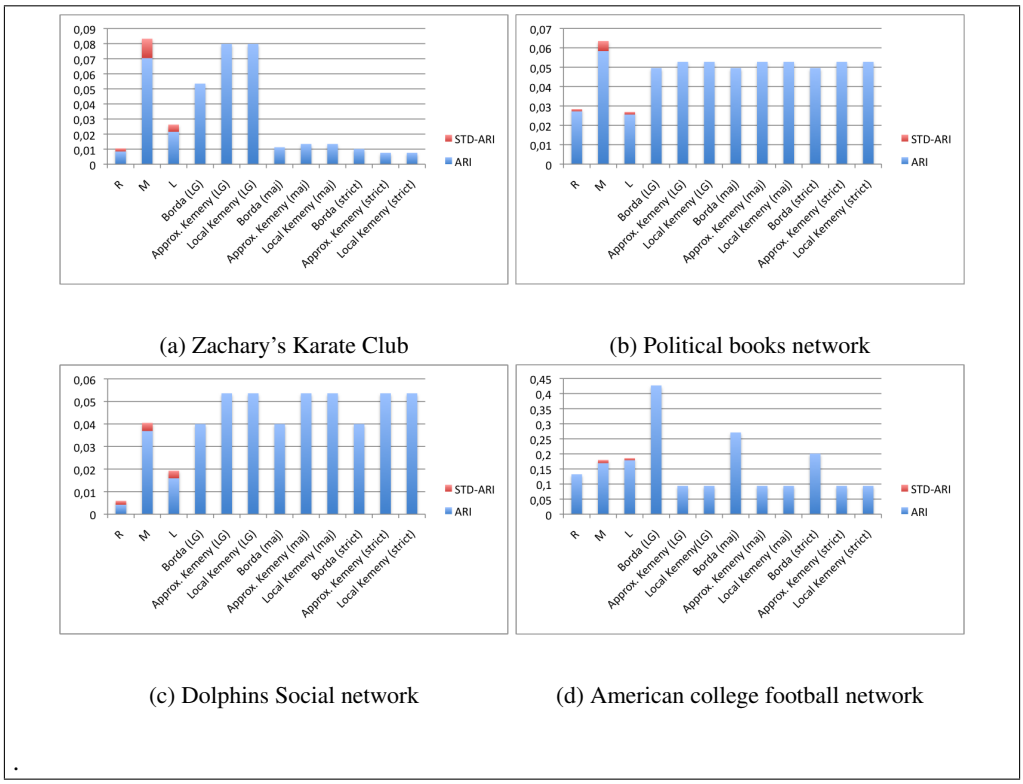


FIG. 2: Compartive results in terms of NMI of basic and ensemble ranking approaches

ensemble ranking approaches that generally gives better results that unsupervised ensemble ranking (Pujari et Kanawati, 2012).

6 Conclusion

In this preliminary work, we have explored applying ensemble ranking approaches to enhance the performances of classical local modularity greedy optimisation algorithms for ego-centered communities identification in complex networks. A straightforward modification of the basic greedy optimisation algorithm is proposed and evaluated on benchmark real networks. Obtained results argue for the validity of the approach. Further experiments are required in order to confirm these first learned lessons. Other approaches of local modularities combination can be imagined ; including a basic combine & rank approach, and ensemble clustering approaches (Strehl et Ghosh, 2003). Following the first approach, we start by computing the quality of each candidate node in the set S according to each applied local modularity and then a global quality value is inferred for all computed qualities (after normalization). Nodes are then ranked according to this averaged quality value. Following an ensemble clustering approach local communities for each node, computed by using different local modularities, are merged to produce the final local community (Seifi, 2012; Dahlin et Svenson, 2013). Compa-



(a) Zachary's Karate Club

(b) Political books network

(c) Dolphins Social network

(d) American college football network

.

FIG. 3: Comparative results in terms of ARI of basic and ensemble ranking approaches

risons with these approaches as well as validation on artificial networks generated by the LFR generator make part of our futur work.

Références

Aggarwal, C. C. et C. K. Reddy (Eds.) (2014). *Data Clustering : Algorithms and Applications*. CRC Press.

Bagrow, J. P. (2008). Evaluating local community methods in networks. *J. Stat. Mech. 2008*(5), P05001.

Bagrow, J. P. et E. M. Bollt (2005). A local method for detecting communities. *Phys. Rev. E 72*, 046108.

Borda, J. C. (1781). Mémoire sur les élections au scrutin. *Comptes rendus de l'Académie des sciences, traduit par Alfred de Grazia comme Mathematical Derivation of a election system , Isis, vol 44, pp 42-51*.

Brandes, U., D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, et D. Wagner (2008). On modularity clustering. *IEEE Trans. Knowl. Data Eng. 20*(2), 172–188.

Chen, J., O. R. Zaïane, et R. Goebel (2009). Local community identification in social networks. In *ASONAM*, pp. 237–242.

Chevaleyre, Y., U. Endriss, J. Lang, et N. Maudet (2007). A short introduction to computational social choice. *SOFSEM 2007 : Theory and Practice of Computer Science*, 51–69.

Clauset, A. (2005). Finding local community structure in networks. *Physical Review E*.

Dahlin, J. et P. Svenson (2013). Ensemble approaches for improving community detection methods. *CoRR abs/1309.0242*.

Danisch, M., J.-L. Guillaume, et B. L. Grand (2013). Unfolding ego-centered community structures with a similarity approach. In *4th Workshop on Complex Networks (CompleNet 2013)*.

Dwork, C., R. Kumar, M. Naor, et D. Sivakumar (2001). Rank aggregation methods for the Web. In *WWW*, pp. 613–622.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports 486*(3-5), 75–174.

Fred, A. L. N. et A. K. Jain (2003). Robust data clustering. In *CVPR (2)*, pp. 128–136. IEEE Computer Society.

Girvan, M. et M. E. J. Newman (2002). Community structure in social and biological networks. *PNAS 99*(12), 7821Ð7826.

Hubert, L. et P. Arabie (1985). Comparing partitions. *Journal of classification 2*(1), 192–218. EN

Kemeny, J. G. (1959). Mathematics without Numbers. *Daedalus 88*, 571–591.

Krebs, V. Political books network. http ://www.orgnet.com/.

Lancichinetti, A. et F. Radicchi (2008). Benchmark graphs for testing community detection algorithms. *Physical Review E Physical Review E*(4), 046110.

Lapata, M. (2006). Automatic Evaluation of Information Ordering : Kendallâs Tau. *Computational Linguistics* (December 2005).

Lim, K. H. et A. Datta (2013). A seed-centric community detection algorithm based on an expanding ring search. In *Proceedings of the First Australasian Web Conference (AWC 2013)*, Adelaide, Australia, pp. 21–25.

Luo, F., J. Z. Wang, et E. Promislow (2008). Exploring local community structures in large networks. *Web Intelligence and Agent Systems 6*(4), 387–400.

Lusseau, D., K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, et S. M. Dawson (2003). The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology 54*, 396–405.

Ma, L., H. Huang, Q. He, K. Chiew, J. Wu, et Y. Che (2013). Gmac : A seed-insensitive approach to local community detection. In L. Bellatreche et M. K. Mohania (Eds.), *DaWaK*, Volume 8057 of *Lecture Notes in Computer Science*, pp. 297–308. Springer.

Meila, M. (2003). Comparing clusterings by the variation of information. In B. Schölkopf et M. K. Warmuth (Eds.), *COLT*, Volume 2777 of *Lecture Notes in Computer Science*, pp. 173–187. Springer.

Melville, P., K. Subbian, E. Meliksetian, et C. Perlich (2010). A Predictive Perspective on Measures of Influence in Networks. *5th Annual Machine Learning Symposium*, 1–5.

Ngonmang, B., M. Tchuente, et E. Viennet (2012). Local community identification in social networks. *Parallel Processing Letters 22*(1).

Nguyen, X. V., J. Epps, et J. Bailey (2009). Information theoretic measures for clusterings comparison : is a correction for chance necessary ? In A. P. Danyluk, L. Bottou, et M. L. Littman (Eds.), *ICML*, Volume 382 of *ACM International Conference Proceeding Series*, pp. 135. ACM.

Palla, G., I. Derônyi, I. Farkas, et T. Vicsek (2005). Uncovering the overlapping modular structure of protein interaction networks. *FEBS Journal 272*, 434.

Pujari, M. et R. Kanawati (2012). Supervised rank aggregation approach for link prediction in complex networks. In A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich, et S. Staab (Eds.), *WWW (Companion Volume)*, pp. 1189–1196. ACM.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association 66*, 846–850.

Seifi, M. (2012). *Cœurs stables de communautés dans les graphes de terrain*. Ph. D. thesis, Université Pierre et marie Curie (paris 6).

Shi, C., Y. Cai, D. Fu, Y. Dong, et B. Wu (2013). A link clustering based overlapping community detection algorithm. *Data Knowl. Eng. 87*, 394–404.

Strehl, A. et J. Ghosh (2003). Cluster ensembles : a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research 3*, 583–617.

Tang, L. et H. Liu (2010). *Community Detection and Mining in Social Media*. Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers.

Wei, F., W. Li, et S. Liu (2010). irank : A rank-learn-combine framework for unsupervised ensemble ranking. *JASIST 61*(6), 1232–1243.

Whang, J. J., D. F. Gleich, et I. S. Dhillon (2013). Overlapping community detection using seed set expansion. In Q. He, A. Iyengar, W. Nejdl, J. Pei, et R. Rastogi (Eds.), *CIKM*, pp. 2099–2108. ACM.

Wu, Y., H. Huang, Z. Hao, et F. Chen (2012). Local community detection using link similarity. *J. Comput. Sci. Technol. 27*(6), 1261–1268.

Xie, J., S. Kelley, et B. K. Szymanski (2013). Overlapping community detection in networks : The state-of-the-art and comparative study. *ACM Comput. Surv. 45*(4), 43.

Zachary., W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research 33*, 452–473.

Zhang, T. et B. Wu (2012). A method for local community detection by finding core nodes. In *ASONAM*, pp. 1171–1176. IEEE Computer Society.

Summary

In this paper we present a new approach for efficiently identifying ego-centered communities in complex networks. Most existing approaches are based on applying a greedy optimisation process guided by a given objective function. Different objective functions has been proposed in the scientific literature, each capturing some specific topological feature of desired

communities. In this work, we propose to apply social choice algorithms in order to combine different objective functions. We compare the proposed approach with state of the art algorithms. Results of experiments on benchmark networks show the relevancy of our approach.

Identification des communautés au sein des réseaux sociaux par l'Analyse Formelle de Concept

Sid Ali Selmane*, Fadila Bentayeb*, Rokia Missaoui**, Omar Boussaid*

*Laboratoire Eric Université Lyon 2, France
Prénom.Nom@univ-lyon2.fr
**Laboratoire LARIM, Canada

Résumé. L'étude des structures de communautés dans les réseaux sociaux (RS) est devenue un vrai challenge ayant de nombreuses applications dans des domaines de recherche multiples. En Informatique, les RS ont été principalement étudiés par deux familles d'approches, celles qui se basent sur la théorie des graphes et celles basées sur l'AFC (Analyse Formelle des Concepts). En effet, plusieurs approches existent en AFC pour la détection de communautés. Cependant, elles n'exploitent que partiellement les RS en considérant seulement une partie des acteurs du RS ayant certaines propriétés. Pour pallier à ce problème, nous proposons dans cet article une nouvelle méthode de détection de communautés en considérant l'ensemble des acteurs du RS. Notre approche est validée à travers des expérimentations sur des RS réels connus dans le domaine et sur des RS issus d'un benchmark de réseaux synthétiques.

1 Introduction

L'identification des communautés s'inscrit dans la problématique générale de l'algorithmique des grands graphes Fortunato (2010). Elle permet d'aider à la compréhension des structures et du fonctionnement de ces derniers. L'analyse de la structure des grands graphes permet de mettre en évidence des communautés qui sont des sous-ensembles de nœuds très fortement liés par rapport au reste des nœuds du graphe. L'analyse de la structure des communautés dans les réseaux a généré diverses approches de partitionnement de graphes issues de la théorie des graphes ainsi que diverses approches de regroupement hiérarchique dans divers domaines dont les sciences sociales et physiques. L'intérêt marqué par la recherche dans ce domaine provient du fait de cibler ces communautés pour un besoin en marketing (diffusion publicitaire), en politique (électorat communautaire), en recherche (diffusion d'articles scientifiques dans une communauté de chercheurs), etc.

L'objectif principal de ce travail est de s'intéresser à une famille de méthodes de détection de communautés à savoir celle basée sur l'AFC Missaoui (2013) en proposant une nouvelle approche qui permet d'affiner la méthode proposée par Falzon (2000) pour la détection de communautés. En effet, l'auteur dans Falzon (2000) se base sur l'AFC pour détecter des communautés dans les RS mais répond partiellement à cette problématique en considérant seulement une partie du RS traité. Dans ce papier, nous proposons d'utiliser une combinaison

des outils et approches de l'AFC et les notions issues de la théorie des graphes pour répondre d'une manière complète au problème de la détection de communautés tout en considérant l'ensemble des acteurs du RS. Notre approche construit d'abord un contexte formel représentant le RS et détermine les communautés partielles. Ensuite, nous affectons les nœuds isolés aux communautés déjà construites tout en maximisant la fonction de modularité que nous avons proposée.

L'article est organisé de la manière suivante. La section 2 présente un état de l'art sur les travaux relatifs à la détection de communautés dans les réseaux. La section 3 porte sur la formalisation de la problématique de détection de communautés selon les approches basées sur l'AFC à savoir celles proposées par Freeman (1996a) et Falzon (2000). Ensuite, dans la section 4, nous développons notre approche et nous déroulons l'algorithme proposé sur un exemple de RS connu du domaine (Le club de karaté de *Zachary*). Nous présentons également les différentes expérimentations que nous avons menées ainsi que les résultats que nous avons obtenus. Nous avons par ailleurs défini une mesure de qualité adaptée au problème d'identification de communautés (issue des indicateurs rappel et précision Salton et Buckley (1997) pour montrer la pertinence des communautés obtenues. Nous concluons et présentons les perspectives de nos travaux dans la section 5.

2 Les communautés dans les réseaux

Les RS sont des groupes d'individus (entités sociales) connectés par des liens sociaux. Les relations entre les individus dépendent du contexte, elles peuvent être des relations d'amitié dans le cas d'un réseau de connaissances, des relations de citations dans un réseau de publication scientifique, des liens de connexions physiques ou logiques dans un réseau informatique, etc. L'analyse des réseaux sociaux couvre un ensemble de problèmes dont les principaux sont l'identification des communautés et leur évolution, l'étude de la dynamique d'un réseau, l'identification des rôles des nœuds et des communautés qui forment un RS, l'étude et la prédiction de liens et la recommandation et cela dans plusieurs domaines d'applications : propagation et recherche d'information dans le web Tummarello et Morbidoni (2008), en sécurité McDaniel et al. (2006), en biologie Girvan et Newman (2002), etc. Les graphes sociaux sont caractérisés par certaines zones denses. Les individus de ces zones denses ont plus de liens entre eux qu'avec le reste du graphe social Fortunato et Barthélemy (2007), Newman (2004). Ces zones denses sont appelées des communautés. La difficulté est de détecter les communautés dans les réseaux, sans connaître ni leur nombre ni leur taille.

D'autres alternatives à la théorie des graphes traitent du problème de détection de communautés. Dans ce papier, nous nous intéressons plus particulièrement à l'AFC, une technique d'extraction des connaissances basée sur la théorie des concepts. Freeman (1996a) fût le premier à utiliser l'AFC pour la découverte de communautés et d'individus importants dans des RS en exploitant le notion de chevauchement des cliques maximales (CM) dans un treillis de Galois qu'il valide sur des réseaux de petite taille issus du monde réel. L'incovénient de cette méthode est l'élimination des acteurs appartenant à des cliques intermédiaires lors du processus de détection. Falzon (2000) améliore cette approche et propose de déterminer des groupes à chaque niveau du treillis en prenant en considération tous les individus appartenant aux CM sans éliminer ceux appartenant aux cliques intermédiaires. Toutefois, cette méthode ne considère pas les individus qui appartiennent au graphe du RS et n'appartenant pas à l'ensemble

des CM. Dans les RS un grand nombre d'individus n'appartient pas à des CM. Partant de ce constat, nous proposons dans cet article une nouvelle méthode de détection de communautés qui améliore l'approche de *Falzon*, en considérant l'ensemble des individus du RS. Autrement dit, aucun acteur du RS n'est éliminé lors du processus de détection proposé. Au final, chaque acteur sera affecté à une communauté. Pour cela nous introduisons une fonction de modularité inspirée de Newman (2004) affinant ainsi le résultat des communautés obtenues.

3 Les communautés dans les approches basées sur l'AFC

L'analyse formelle de concepts (AFC) est un formalisme de représentation et d'extraction de connaissances fondé sur les notions de concepts et de treillis de concepts (Galois). L'AFC a été exploitée avec succès dans plusieurs domaines en informatique tels le génie logiciel, les bases et entrepôts de données, l'extraction et la gestion de la connaissance et dans plusieurs applications du monde réel comme la médecine, la psychologie, la linguistique et la sociologie. Dans cette section nous présentons une formalisation du problème d'identification des communautés dans les approches basées sur l'AFC . Ensuite, nous présentons les méthodes de Freeman (1996a) et Falzon (2000) pour montrer leur limites.

3.1 Identification des communautés dans les approches basées sur l'AFC

Nous présentons dans cette section les différents concepts nécessaires pour l'identification des communautés dans les RS.

3.1.1 Définitions

- Graphe. Soit $G = (V, E)$ un graphe représentant un réseau social où :
  - V est l'ensemble des acteurs (nœuds) $\{x_i\}_{i=1}^{n}$ du réseau social et $n = |V|$ est le nombre de nœuds dans G.
  - E est l'ensemble des liens sociaux entre les acteurs et $m = |E|$ est le nombre d'arêtes dans G.
- Clique. Une clique d'un graphe est un sous ensemble de sommets tous en relation deux à deux. Elle définit un sous graphe à n sommets et $n(n-1)/2$ arêtes.
- Clique Maximale (CM). une clique est dite maximale si elle n'est pas contenu dans une clique de taille plus grande. La recherche de CM est connue pour être un problème NP-Complet mais les recherches récentes ont permis de réduire sa complexité par des heuristiques ou en proposant la parallélisation de leur extraction San Segundo et al. (2011).
- Contexte formel. Soit $F = (V, C, I)$ le contexte formel associant les CM d'un graphe à l'ensemble de ses acteurs V où :
  - C est l'ensembles des CM $\{C_j\}_{j=1}^{k}$ extraites du graphe G.
  - I est la relation binaire qui lie les ensemble V et C. Autrement dit, si un acteur x_i appartient à une CM C_j, $I(x_i, C_j) = 1$ sinon C_j, $I(x_i, C_j) = 0$.

3.1.2 Principe de détection de communautés dans les RS selon l'AFC

Pour identifier des communautés dans les RS, l'AFC exploite le treillis de Galois construit à partir du contexte $F = (V, C, I)$ pour réaliser un clustering du treillis qui représente des groupes de nœuds partageant les mêmes propriétés entre eux.

Dans ce qui suit, nous illustrons les définitions par un exemple de RS représenté par un graphe $G(V, E)$, de 15 acteurs $V = \{1, 2, 3, ..., 15\}$ et 32 arrêtes, de ce graphe nous extrayons 4 CM $C = \{a, b, c, d\}$. Ensuite nous construisons le contexte formel $\mathbb{K} := (V, C, I)$ (Figure 1), composé de trois ensembles V, C et I relation binaire qui lie les ensembles V et C. Cette relation est vrai si un acteur dans V appartient à une CM C. Cet exemple sert à expliquer la formalisation du problème.

$\mathbb{K}$	a	b	c	d
1	1	0	0	0
2	1	1	0	0
3	1	0	0	1
4	1	0	1	1
5	0	0	1	0
6	0	1	0	1
7	0	1	1	1
8	0	0	0	1
9	0	1	0	1
10	0	0	1	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0

FIG. 1 – *Contexte formel* $\mathbb{K} := (V, C, I)$.

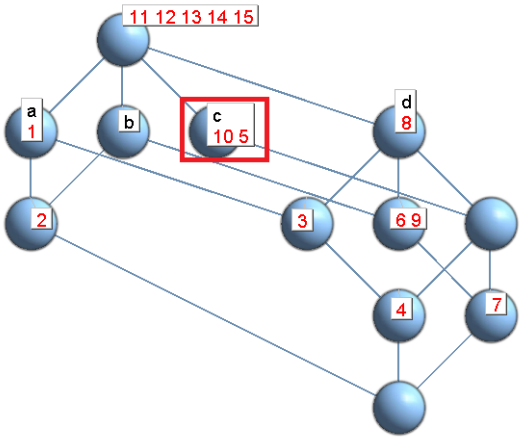


FIG. 2 – $\mathfrak{T}(\mathbb{K})$ *Treillis de Galois du contexte* $\mathbb{K}$.

Un contexte formel est un triplet $\mathbb{K} := (G, M, I)$ où G est un ensemble d'objets, M un ensemble d'attributs et I une relation binaire qui lie les ensembles G et M, $I \subseteq G \times M$. Pour $A \subseteq G$ et $B \subseteq M$, deux sous-ensembles $A' \subseteq M$ et $B' \subseteq G$ sont définis respectivement comme un ensemble d'attributs communs aux objets dans A et l'ensemble d'objets qui partagent tout attribut dans B. Formellement, la dérivation notée $'$ est définie comme suit :

$$A' := \{a \in M \mid oIa \; \forall o \in A\}\text{(intension)} \quad \text{et} \quad B' := \{o \in G \mid oIa \; \forall a \in B\}\text{(extension)}.$$

Cette proposition définit une paire de correspondance $('\,,'\,)$ entre l'ensemble des parties de G et l'ensemble des parties de M, qui est une correspondance de Galois. Les opérateurs de la fermeture induite (dans G et M) sont notés $''$. Dans notre exemple, l'ensemble des acteurs V représente l'ensemble des objets G et l'ensemble des CM C représente l'ensemble des attributs.

Exemple : soit $A_1 = \{6, 7\}, A_1 \subset G \Rightarrow (A_1)' = \{b, d\}$ et soit $B_1 = \{c, d\}, M_1 \subset M \Rightarrow (B_1)' = \{4, 7\}$

Un *concept formel* (fermé, rectangle) cf est une paire (A, B) avec $A \subseteq G$, $B \subseteq M$, $A = B'$ et $B = A'$. L'ensemble A, qu'on notera Ext(cf), est appelé *extension* de cf tandis que B est son *intention*, qu'on notera Int(cf). Un concept formel correspond à un rectangle maximal dans un contexte formel.

L'ensemble $\mathfrak{B}(\mathbb{K})$ de tous les concepts est formé comme suit :

$$\mathfrak{B}(\mathbb{K}) = \{(A, B) \in (G', M') \setminus A = B' \text{et} B = A'.\}$$

Exemple : soit $A_1 = \{6, 7\}, A_1 \subset G \Rightarrow (G_1)' = \{b, d\} \Rightarrow ((A_1)')' = (\{b, d\})' = \{6, 7, 9\}$ donc l'ensemble $A_1 \notin \mathfrak{B}(\mathbb{K})$ car la définition n'est pas satisfaite.

L'ensemble $\mathfrak{B}(\mathbb{K})$ de tous les concepts du contexte $\mathbb{K}$ ordonnés partiellement par : $(X_1, Y_1) \leq (X_2, Y_2) \Leftrightarrow X_1 \subseteq X_2$ constitue un treillis complet, appelé treillis de concepts de $\mathbb{K}$ et noté $\mathfrak{T}(\mathbb{K})$. La figure 2 montre $\mathfrak{T}(\mathbb{K})$ le treillis de concepts obtenu à partir du contexte formel $\mathbb{K}$ de la figure 1. L'étiquetage du treillis de la figure 2 est réduit au niveau des attributs de sorte que l'intention d'un concept (nœud) n est donnée par l'union des attributs apparaissant dans le nœud n ainsi que ceux apparaissant dans les concepts qui sont plus petits que n.

Par exemple, le nœud encadré en rouge avec l'étiquette $c - 10, 5$ représente le concept $(\{10, 4, 5, 7\}, \{\mathbf{c}\})$ (Figure 2).

Le haut du treillis (supremum) (11, 12, 13, 14, 15) montre qu'aucun de ces acteurs n'appartient à une CM dans Freeman (1996b) et Falzon (2000) les auteurs ne les considèrent pas et ne seront assignés à aucun groupe après le processus d'identification de communautés, le bas du treillis (infimum) est vide montre qu'aucun acteur n'appartient à toutes les CM par contre les acteurs 2, 4 et 7 parents de l'infimum seront considérés comme acteurs centraux dans ce réseau Freeman (1996b),Falzon (2000).

Le chevauchement de CM dans un treillis de Galois veut dire que deux CM dans le treillis partagent au moins un acteur en commun (autrement dit, si les CM se rejoignent uniquement en l'infimum elles ne se chevauchent pas). Le chevauchement est réflexif, transitif et symétrique. Ainsi, le chevauchement est une relation d'équivalence. Graphiquement, on peut remarquer le chevauchement entre deux nœuds du niveau *k* s'ils se rejoignent au niveau *k+1* (figure 2).

3.2 Méthodes de Freeman et Falzon pour la détection de communautés

La méthode proposée par Falzon (2000) repose sur le même fondement théorique de représentation du RS que celui proposé par Freeman (1996a) qui associe le concept formel de la CM au concept formel du treillis de Galois pour la détermination de communautés. Pour détécter des communautés, Freeman (1996a) se base notamment sur la notion de chevauchement de CM dans le treillis de Galois. Freeman (1996b) détermine à partir du treillis de Galois $\mathfrak{T}(\mathbb{K})$, un ensemble de CM dont au moins deux chemins allant de sa position courante dans le treillis vers l'infimum ne sont pas de même longueur (figure 3 chemins pointillés.) qu'il appelle cliques intermédiaires. Ensuite il procède à l'élimination des arêtes partant de ces nœuds pour obtenir des groupes disjoints (figure 4).

Dans les grands treillis plusieurs nœuds appartiennent à des cliques intermédiaires et sont éliminés lors du processus de détection proposé par *Freeman*. Partant de cette limite, Falzon (2000) propose une nouvelle approche qui n'élimine pas ce types de nœuds et la justifie à travers l'observation d'une propriété du treillis de CM. Falzon (2000) suppose que la structure du groupe évolue parallèlement au modèle de chevauchement dans les couches du treillis

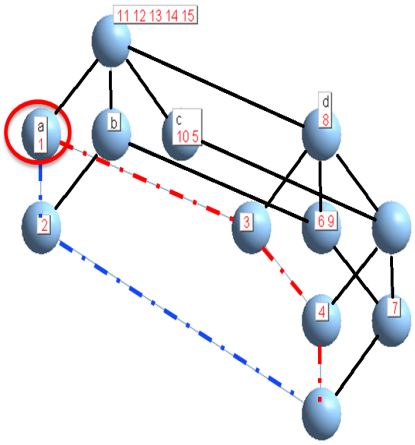


FIG. 3 – *Cliques intermédiaires*

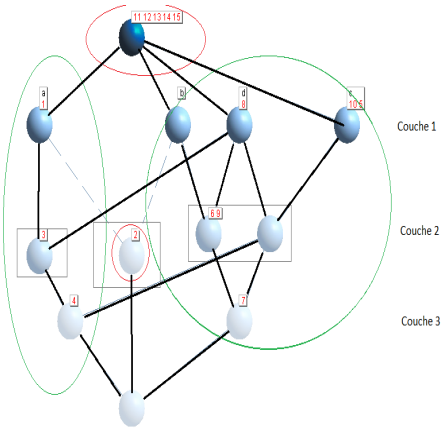


FIG. 4 – *Groupes obtenus aprés éliminations des arêtes*

en dessous de la première couche qui représente l'ensemble des CM. Si nous prenons à titre d'exemple le treillis de la figure 4, nous remarquons que les nœuds dans la *couche 2* peuvent être parfaitement divisées en trois groupes disjoints (représentés par des rectangles). De cette façon, nous obtenons essentiellement la même structure que nous aurions eu en supprimant les arrêtes des cliques intermédiaires. Dans les grands réseaux, on s'attendrait à une structure de groupe plus raffiné chaque fois que l'on descend dans les couches du treillis. La complexité de la construction du treillis de Galois est exponentiel. Cependant Falzon (2000) propose une méthode de construction qui ne calcule qu'en partie ce dernier de manière polynomiale et qui classe tous les acteurs des CM. Pour la construction du treillis, il prend en entrée l'ensemble des CM L_1 puis construit pour chaque niveau (k) un ensemble L_k créé à partir des intersections des paires de nœuds du niveau $k-1$. Ensuite, une liste $LS[k]$ d'ensembles de nœuds pour chaque couche de niveau (k) est créée. Cette liste contient les nœuds dont les acteurs n'apparaissent pas aux niveaux supérieurs à k. Enfin, une comparaison de chaque paire de couches adjacentes du treillis est établie pour déterminer si elles ont des ensembles de nœuds communs, les nœuds communs de la couche supérieure sont éliminés. L'algorithme 1 de la structure de groupes proposé par Falzon (2000) appelle la fonction chevauchement nœud o plusieurs fois. Cette fonction est une extension aux nœuds de la définition de chevauchement de CM, son appel s'effectue une fois pour chaque couche dans le réseau lors du processus de détection de groupes.

La création des groupes G_k se fait d'abord par la génération des groupes au niveau 1. Ces groupes sont formés en appliquant la fonction de chevauchement sur les nœuds du niveau 1, puis les groupes suivants (avec $k > 1$) sont formés en appelant la fonction de chevauchement sur les nœuds du niveau k et les nœuds contenus dans les listes $LS[1]$ jusqu'à $LS[k-1]$.

Falzon (2000) affirme que la méthode de détection de groupes de *Freeman* est bonne pour les contextes de petite taille dont le treillis de CM associé n'excède pas trois niveaux. Au delà de trois niveau le nombre d'acteurs appartenant aux cliques intermédiaires devient important et par conséquent ils sont éliminés de l'ensemble des groupes détectés. Falzon (2000) répond

Algorithme 1 : L'algorithme de la structure de groupe selon la méthode de *Falzon*

1: $LS[0] = \emptyset$
2: **for** $i := 0$ to $maxplayer - 1$ **do**
3: $L := L_{i+1} \cup LS[1] \cup LS[2] \cup ... \cup LS[i]$;
4: $k := 1$;
5: **while** L non vide ; **do**
6: Soit n le premier ensemble nœud de L ;
7: $GS := n$;
8: Déterminer tous les ensembles nœud n_j tel que $(n, n_j) \in o$ (fonction de chevauchement) ;
9: $G_k := \cup n_j$; $GS := n_j$;
10: $L := L - GS$;
11: Ajouter G_k à la liste de groupes pour la couche i ;
12: $k++$
13: $i := i + 1$;

à cette perte d'information pertinente en proposant une méthode qui n'élimine pas les acteurs appartenant aux cliques intermédiaires et qui détermine des groupes à chaque niveau du treillis, mais toujours à partir du même treillis de CM que *Freeman* propose de construire en premier. Or dans les graphes de RS un grand nombre d'acteurs n'appartient pas à des CM. L'exemple de RS construit par Falzon (2000) pour illustrer ces algorithmes comprend au départ 97 acteurs, ensuite 18 sont éliminés car ils n'appartiennent à aucune CM, ce qui représente 19 % du RS. Partant de cette limite qui ne prend pas en considération l'ensemble des acteurs du RS (acteurs du supremum du treillis de CM), nous proposons une nouvelle approche qui prend en compte l'ensemble des acteurs d'un RS lors du processus de détection. Nous affectons chaque acteur du réseau à au moins une communauté.

4 Approche proposée

Les approches de détection de communautés basées sur l'AFC permettent de répondre partiellement à la problématique. En effet, elles prennent en considération seulement une partie du RS sur laquelle elles appliquent leur approches d'identification de communautés. Partant de ce constat, l'idée de trouver une approche qui considère l'ensemble des acteurs du RS est très intéressante. Effectivement, dans le domaine de détection de communautés, certains acteurs peuvent jouer des rôles importants malgré qu'ils n'entretiennent pas beaucoup de liens au sein du RS lui même (autrement dit, ils n'appartiennent pas à des CM). Par exemple, dans une organisation terroriste, les personnes qui exécutent des attentats entretiennent peu de contacts avec les autres terroristes d'un même groupe, ils attendent juste un ordre d'une seule personne qui leur assigne une mission. Ainsi un seul contact permet au terroriste en question de commettre son acte. La modélisation de cette organisation en RS représentera ce contact par un seul lien et donc cet acteur n'appartiendrait à aucune CM. Dans les approches basées sur l'AFC décrites précédemment, ces acteurs n'appartiendront à aucune CM et de ce fait ne seront pas considérés lors du processus de détection. Par contre les éléments chargés de la logistique, de la préparation de cette organisation auront beaucoup de contacts entre eux et appartiendront à des

CM alors que ces éléments ne vont pas agir concrètement. D'où l'importance de considérer des éléments isolés lors du processus d'identification de communautés car non seulement un réseau de terroristes peut être démantelé mais aussi un attentat peut être évité en reliant ces éléments entretenant peu de contacts avec les groupes détectés.

4.1 Principe général

Dans ce qui suit, nous présentons la démarche que nous proposons illustrée par un exemple tout en se basant sur les définitions formelles de la section 3.2. L'identification de communautés est un tâche difficile à accomplir car le nombre de communautés et leurs tailles (le nombre de nœuds qu'elles contiennent) ne sont pas connus à priori. Tout en se basant sur le formalisme défini dans la section 3, notre approche permet d'apporter une solution qui combine l'approche de Falzon (2000) pour la détection de communautés avec une notion de la théorie des graphes qui est l'adaptation de la fonction de modularité de Newman (2004). Le processus d'identification de communautés que nous proposons s'effectue en deux étapes. Dans la première étape nous déterminons les groupes de niveaux G_k en appliquant l'algorithme de Falzon (2000) sur une partie du RS (les acteurs appartenant à des CM), puis dans la deuxième étape nous construisons notre contexte formel $K^* = (G(V, E), G_k, L_1)$ constitué du graphe du RS $G(V, E)$, des groupes de niveaux G_k et l'ensemble des nœuds L_1 représentant les CM au niveau 1 du treillis. Puis, nous déterminons l'ensemble des nœuds N_a du supremum du treillis, c.à.d, les acteurs qui appartiennent au graphe $G(V, E)$ et n'appartenant pas à l'ensemble des CM L_1. Partant de l'hypothèse qu'un acteur appartient à un groupe avec lequel il existe au moins un lien avec l'un des acteurs de ce groupe, pour chaque nœud n_i appartenant à l'ensemble N_a, nous générons des conteneurs $B(n_i)$ à partir de l'ensemble des arêtes E et l'ensemble des groupes G_k qui contiennent l'ensemble des paires (n_j, G_k) représentant les nœuds et les groupes avec lesquels le nœud n_i a un lien. Une fois ce conteneur construit, nous vérifions si le nœud n_i a des liens uniquement avec des nœuds appartenant au même groupe ou avec des nœuds qui appartiennent à différents groupes. Dans le premier cas le nœud n_i sera assigné au groupe auquel il a un ou plusieurs liens. Dans le deuxième cas nous calculons la fonction de modularité adaptée par rapport aux différents groupes auxquels il a des liens, c.à.d en considérant le nœud n_i appartenant à chacun des groupes G_k avec lesquels il partage un lien nous calculons et comparons les différentes valeurs de $Q(G_k, n_i)$ pour la maximiser, elle est donnée par :

$$Q(G_k, n_i) = \sum_j (e_{jj}(G_k) - a_j^2(G_k^*))$$

où $e(G_k)_{jj}$ est la proportion d'arêtes à l'intérieur des groupes (nombre d'arêtes dans le groupe G_k en prenant en considération le nœud n_i divisé par le nombre total d'arêtes dans le graphe), $a_j(G_k^*) = \sum_j (e_{jj}(G_k^*))$ est la proportion d'arêtes attendue dans le graphe de G en assignant le nœud n_i au groupe G_k^*. Le nœud n_i sera assigné au groupe dont $Q(G_k, n_i)$ est maximale.

Pour évaluer la précision des partitionnements que nous obtenons à travers chaque niveau nous avons adapté les mesures de Rappel et de Précision dans le cas des multiclasses à notre approches d'identification de communautés, soit $P = (G_1, G_2, ..., G_k)$, $\bigcup_{(i=1..k)} G_i = V$ et $\forall i \neq j, G_i \cap G_j = \emptyset$ un partitionnement de V en k communautés. Initialement chaque n œud est attribué à une seule communauté G_j, cette attribution est déterminé soit par des études

(ethnographiques, sociologiques, politiques, etc.) pour le cas des RS issus du monde réel, ou bien obtenu par le modèle de génération pour les RS synthétiques. A travers notre méthode pour chaque niveau du treillis nous obtenons un partitionnement P' que nous comparons avec le partitionnement initial P, $P' = (G'_1, G'_2, ..., G'_l)$, $\bigcup_{(i=1..l)} G'_i = V$ et $\forall i \neq j, G'_i \cap G'_j = \emptyset$.

La précision $\mathbb{P}(P, P')$ et le rappel $\mathbb{R}(P, P')$ du partitionnement sont donnés par les formules suivantes :

$$\mathbb{P}(P, P') = \frac{\sum_{i=1}^{l} \mathbb{P}\{G'_i\}}{l} \quad (1) \qquad \mathbb{R}(P, P') = \frac{\sum_{i=1}^{l} \mathbb{R}\{G'_i\}}{l} \quad (2)$$

La précision $\mathbb{P}\{G'_i\}$ représente le nombre d'individus correctement regroupés dans une communauté G'_i par rapport au nombre d'individus initialement dans la communauté G_i et le rappel $\mathbb{R}\{G'_i\}$ le nombre d'individus correctement regroupés dans une communauté G'_i par rapport au nombre total d'individus dans la communauté G'_i. Ces mesures sont données par les formules suivantes :

$$\mathbb{P}\{G'_i\} = \frac{|G'_i \cap G_i|}{|G'_i|} \quad (3) \qquad \mathbb{R}\{G'_i\} = \frac{|G'_i \cap G_i|}{|G_i|} \quad (4)$$

4.2 Algorithme et exemple illustratif

Algorithme 2 : Pseudo code de l'algorithme de structure des groupe Affinés

1: **Procédure** GROUPS $(G(V, E), G_k, L_1)$
2: **In :** $G(V, E)$ Graphe du réseau social, G_k groupes obtenues à partir du treillis de CM et L_1 l'ensemble des CM
3: **Out :** Ensemble de Groupes affinés G^*_K du graphe du réseau social.
4: $G^*_K = G_k$
5: Déterminer N_a l'ensemble de nœuds n_i du supremmum du treillis, $n_i \in G(V, E)$ et $n_i \notin L_1$.
6: **for all** $n_i \in |N_a|$ **do**
7: Construire les conteneurs $B(n_i)$ à partir des groupes G_k auxquels appartiennent les nœuds voisins de n_i. {$B(n_i)$ est formé de couples (n_i, G_i) pour garder trace des différents G_k groupe auquel n_i à un voisin.}
8: **if** tout les nœuds de $B(n_i)$ appartiennent au même groupe G_i **then**
9: $G^*_i = G_i \cup n_i$ rajouter le nœud n_i au groupe de niveau affiné
10: **else**
11: Calculer la modularité adaptée du nœud n_i pour chaque élément de $B(n_i)$
12: Comparer les valeurs $Q(G_k, n_i)$ pour chaque groupe G_i et affecter le nœud au groupe de plus grande $Q(G_k, n_i)$. $G^*_i = G_i \cup n_i$.
Dans le cas ou les valeurs sont égales nous affecterons au dernier groupe pour lequel on calcul la modularité
13: **Out :** G^*_i

L'approche que nous proposons est traduite à travers le pseudo code de l'algorithme 2 que nous avons proposé. L'algorithme prend en entrée l'ensemble des acteurs, l'ensemble des groupes de niveau et l'ensemble des CM. En sortie nous avons des groupes affinée qui incluent tout les acteurs. La première étape de l'algorithme ligne 5 consiste à déterminer les acteurs

isolés, les lignes 6 à 9 construisent les conteneurs des groupes avec lesquels les nœuds isolés ont un lien, enfin les lignes 10 à 12 calculent la modularité adaptée et affectent les nœuds isolés à leur groupes.

Pour dérouler notre algorithme, nous avons choisi le club de karaté de Zachary (1977). C'est un RS simple et connu dans la littérature. Le réseau de *Zachary* est un RS des membres d'un club de karaté de l'université San Francisco aux états Unis, qui compte 78 membres. $Zachary$ a fait une étude sur les membres du club qui ont des relations d'amitié en dehors du club. Parmi les 78 membres du club, seuls 34 entretiennent des relations d'amitié. Le réseau d'amitié issu de ce club est représenté dans la figure 6. Il est constitué de 34 nœuds représentant les membres du club et 78 liens représentant les amitiés entre les membres. Ce RS contient deux communautés réelles, instructeurs et administrateurs du club séparé par un trait vertical.

A partir de ce graphe et à l'aide d'UCINET6, [1] nous avons déterminé un ensemble de 25 CM qui représentent la première couche du treillis mais qui contient que 32 des 34 acteurs car le nœud 12 et le nœud 10 n'appartiennent à aucune CM. Nous construisons par la suite le contexte $\mathbb{F}(Zachary) = (V, C, I)$ formé par l'ensemble des acteurs, l'ensemble des CM et de la relation d'appartenance. De ce contexte nous obtenons le treillis $\mathfrak{T}(\mathbb{F}(Zachary))$ de CM représenté par des ensemble de parties formant ces différentes couches sachant que la première couche énumère l'ensemble des CM et les couches suivantes mettent en évidence les différentes intersections avec élimination des nœuds communs.

En appliquant l'algorithme 1 sur le treillis $\mathfrak{T}(\mathbb{F}(Zachary))$, nous déterminons les groupes à chaque niveau :

- Groupes au niveau 1 : {{1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34 }}
- Groupes au niveau 2 : { {1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34}}
- Groupes au niveau 3 : {{6, 7, 17}, {9, 15, 16, 19, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32,33, 34 }, {1, 2, 3, 4, 5, 8, 11, 13, 14, 18, 20, 22 }}
- Groupes au niveau 4 : {{1, 2, 3, 4, 5, 8, 11, 13, 14, 18, 20, 22 }, {6, 7, 17}, {9, 15, 16, 19, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34}}

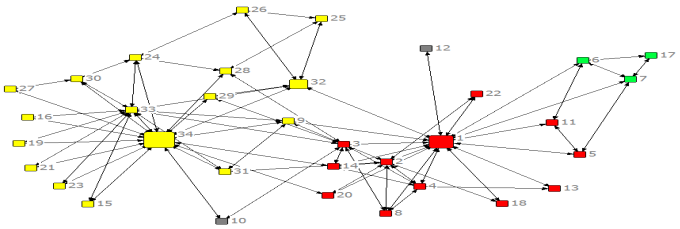


FIG. 5 – *Groupes au niveau 3 et 4 du Club de Karaté de ZACHARY.*

Au final nous obtenons trois groupes distincts (Figure 5). Les différents groupes obtenus sont coloriés en rouges, jaunes et verts. Les acteurs 10 et 12 n'appartiennent à aucune CM ils seront donc affectés à l'un des trois groupes. L'acteur 12 ayant un lien direct avec le groupe 3

1. https ://sites.google.com/site/ucinetsoftware/

en rouge sera affecté à ce groupe. L'acteur 10 ayant deux liens l'un avec le groupe 1 en jaune et l'autre lien avec le groupe 3 en rouge nous calculons alors la fonction de modularité adapté $Q(G_k, n_i)$ pour pouvoir l'affecter (Figure 6).

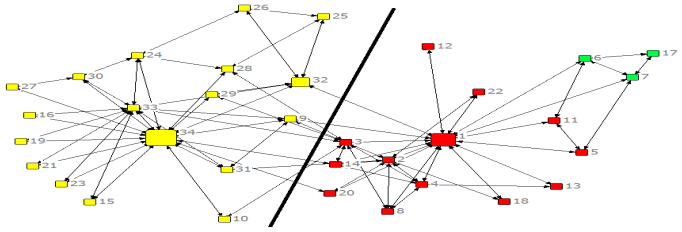


FIG. 6 – *Groupes affinés au niveau 3 et 4 du Club de Karaté de ZACHARY.*

4.3 Expérimentations et discussion

Pour valider notre approche, nous avons considéré plusieurs RS. Tout d'abord nous l'avons testé sur les exemples cités dans les travaux de $Freeman$ et de $Falzon$, ensuite sur des RS issus du monde réel pour montrer que même pour des RS de taille moyenne de nombreux acteurs sont éliminés et ne sont affectés à aucune communauté, prenons l'exemple du RS $Dolphin$ (figure 7) sur les 62 acteurs de ce réseau, 16 acteurs n'appartiennent à aucune CM (26% du RS) et ne sont affectés à aucune communautés par les approches précédentes mais que à travers notre approche nous identifions à quelle communauté ils appartiennent.

Réseau Social	|V|	|E|	|Cliques Max|	Na	nbr de groupes	R	P
Zachary	34	78	25	2	3	0.61	1
Dolphin	62	159	46	16	2	0.95	0.97
Football	115	613	185	10	12	0.87	0.90
Polbooks	105	441	181	22	4	0.70	0.79
L1000n16c	1000	15168	1902	130	16	0.92	1
L5000n32c	5000	75946	9723	679	32	0.87	0.95

FIG. 7 – *Résultats de test*

La figure 7 résume l'ensemble des tests que nous avons menés nous avons pris 4 RS issus du monde réel et trés étudié dans le domaine d'identification de communautés et deux RS synthétiques générés à partir du modèle de Lancichinetti et Fortunato (2009), le plus grand contient 5000 n œuds avec 75946 arêtes. Le but de ces tests est de valider seulement l'approche et montrer sa précision, les tests de passage à l'échelle et de performances (temps d'exécutions) en variant différents paramètres feront l'objet d'un autre travail.

Nous remarquons que notre approche a une bonne précision (i.e., détection correcte des communautés même dans les cas non évidents)et produit des résultats quasi identiques à l'étude ethnographique mené par $Wayne\ Zachary$ (Figure 6). Le premier groupe est le même

que celui retrouvé dans l'étude, par contre nous obtenons un groupe en plus constitué des acteurs 6,7 et 17 qu'on peut intuitivement soit le fusionner avec le groupe en rouge et obtenir les communautés issus de l'étude de $Zachary$, ou bien le considérer comme un nouveau groupe montrant des liens internes très fort. Les communautés que nous obtenons par notre méthode se rapproche le plus à la réalité que d'autre méthodes, plusieurs recherches assignent les acteurs de cette exemple à des communautés dont le nombre varie entre 4 à 7 communautés et des fois pour une même proche en variant des paramètres le nombre de communautés augmente ou diminue.

5 Conclusion

Dans cet article, nous avons proposé une nouvelle méthode de détection de communautés dans les RS qui améliore la méthode proposée par $Falzon$. Notre approche permet de considérer l'ensemble des acteurs d'un RS lors du processus d'identification de communauté tout en se basant sur les fondements théorique de l'AFC. A travers la démarche et l'algorithme proposé nous avons pu montrer comment identifier des communautés avec une bonne précision que nous avons évalué par des tests sur sur des RS réels et synthétiques. Les tests menés ont permis de montrer que la précision de détection est importante tant bien pour des RS issus du monde réel que pour des RS synthétiques. L'une des plus importantes perspectives consiste à faire une étude expérimentale sur des RS synthétiques en considérant différents paramètres de tests à savoir le nombre de nœuds, le degrés des nœuds et la densité des RS. Ces tests nous permettrons d'évaluer les limites de notre méthode pour pouvoir par la suite mener une étude comparative en terme de temps d'exécution et de précision par rapport aux différentes approches qui se basent sur les graphes. Cette démarche peut nous conduire à proposer des approches qui seront appliquées à d'autres types de RS notamment les RS bimodaux qui représentent les interactions entre acteurs et évènements.

Références

Falzon, L. (2000). Determining groups from the clique structure in large social networks. *Social Networks 22*(2), 159–172.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports 486*, 75–174.

Fortunato, S. et M. Barthélemy (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences 104*(1), 36.

Freeman, L. C. (1996a). Cliques, galois lattices, and the structure of human social groups. *Social Networks 18*(3), 173–187.

Freeman, L. C. (1996b). Cliques, galois lattices, and the structure of human social groups. *Social Networks 18*(3), 173–187.

Girvan, M. et M. E. J. Newman (2002). Community structure in social and biological networks. *PNAS 99*(12), 7821–7826.

Lancichinetti, A. et S. Fortunato (2009). Community detection algorithms : a comparative analysis. *Physical review E 80*(5), 056117.

McDaniel, P. D., S. Sen, O. Spatscheck, J. E. van der Merwe, W. Aiello, et C. R. Kalmanek (2006). Enterprise security : A community of interest based approach. In *NDSS*. The Internet Society.

Missaoui, R. (2013). Analyse de réseaux sociaux par l'analyse formelle de concepts. In *EGC*, pp. 3–4.

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E 69*(066133).

Salton, G. et C. Buckley (1997). Improving retrieval performance by relevance feedback. *Readings in information retrieval 24*, 5.

San Segundo, P., D. Rodríguez-Losada, et A. Jiménez (2011). An exact bit-parallel algorithm for the maximum clique problem. *Computers & Operations Research 38*(2), 571–581.

Tummarello, G. et C. Morbidoni (2008). The dbin platform : A complete environment for semanticweb communities. *Web Semantics : Science, Services and Agents on the World Wide Web 6*(4).

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 452–473.

Summary

The study of community structure of SN (social networks) became a real challenge with many applications in multiple research areas. In Computer Science, SN have been studied mainly by two families of approaches, those based on graph theory and those based on the FCA (Formal Concept Analysis). Several approaches exist in FCA for cmmunity detection however they partially exploit SN by considering only a part of the SN actors who share certain properties. This paper proposes to improve a partial detection method by considering all actors of the SN. Our approach is validated through experiments on real known SN in the field and a synthetic benchmark networks.

Maximal connected frequent subgraph mining

Nour el islem Karabadji*, Hassina Seridi*

* Electronic Document Management Laboratory (LabGED),
Badji Mokhtar-Annaba University, P.O. Box 12, 23000 Annaba, Algeria,
{Karabadji, Seridi}@labged.net.

Abstract. While the maximal frequent subset and connected tree mining problems for set and tree datasets can be solved in polynomial delay, it becomes intractable for arbitrary graph databases. Existing approaches have therefore proposed various strategies and restrictions of the search space, but the size of the visited subgraphs before the maximal ones increase the total cost. In this paper,the main contribution is to present the existing maximal connected frequent subgraphs mining algorithms, and to propose a new mining schema for solving the maximal connected frequent subgraphs over unique edge labelled graphs data. The proposed schema reduces the visited subgraphs to the set of the closed ones $\mathcal{C}$. Following this mining schema, subgraphs are generated and their closure operator are computed in polynomial time. This result is due to strongly accessibility of the subgraphs system and the reduction of the subgraph isomorphism into subset relation test. Finally, the closed frequent subgraphs $\mathcal{C}$ are merged to reach the set of the maximals $\mathcal{M}$ over them.

1 Introduction

The problem of frequent substructures mining has been widely studied Agrawal et al. (1993), Gunopulos et al. (2003), Chi et al. (2005). According to the downward closure property, all the substructures of a frequent substructure must be frequent, which renders the enumeration and analysis nearly impossible; if the pattern X is frequent, all of its generalisations patterns (parents) are frequent as well. For instance, if X is a frequent set of n elements then it have 2^n frequents subsets; which is an exponential number. To overcome this problem, closed and maximal substructures mining were proposed, because the set of closed and maximal frequent substructures is much smaller to that of the set of frequent substructures.

According to a threshold $t > 0$, the set of all frequent patterns is noted $\mathcal{F}$, where $\forall x \in \mathcal{F},\ |\mathcal{D}[x]| \geq t$. A frequent pattern $c \in \mathcal{F}$ is called closed if no other pattern $q \in \mathcal{F}$ satisfies $\mathcal{D}[c] = \mathcal{D}[q],\ c \subseteq q$. If a frequent pattern x is included in no other frequent pattern, x is called maximal. The closed frequent pattern set $\mathcal{C}$ contains more complete information of frequent patterns then the maximal ones $\mathcal{M}$; where $\mathcal{C}$ contain the complete support information regarding to its corresponding frequent sub-patterns. The set of closed frequent pattern contains the set of the maximal ones $\mathcal{F} \supseteq \mathcal{C} \supseteq \mathcal{M}$. In the existing algorithms in the literature, the complexity of structures is the main key in the target to classifier the feasibility of those closed $\mathcal{C}$

and maximal $\mathcal{M}$ frequent patterns mining algorithms. The structure complexity has a strongly influences on both the input system (language) and the frequency test queries.

For instances, there are several results on efficient enumeration of the closed sets $\mathcal{C}$ for the case that the underlying set system is finite and closed under intersection Ganter and Reuter (1991), like itemsets set system. Also, the maximal frequent itemset case, where the underlying set system is a lattice system; Dualize and Advance algorithm (first introduced in Gunopulos et al Gunopulos et al. (1997)) computation time comes close to lower bounds for the problem of finnding $\mathcal{M}$. This algorithm is an efficient enumeration algorithm of maximal frequent patterns $\mathcal{M}$ sets for the case that there is a complimentary subset $\overline{X}$ in the set system for every subset X in this set system. In this case of itemsets the frequency test query is polynomial in the size of the transactional data $\mathcal{D}$. In contrast to the itemsets, connected frequent sub-graphs mining algorithms have worse computation results due to the input subgraphs system and sub-graph isomorphism queries.

The main objective in this paper is to deal with the problem of maximal connected frequent subgraphs mining. According to unique labelled graphs data and assuming that we can not in any way investing in the dualization process, and like the maximal frequent patterns set $\mathcal{M}$ is a subset of the closed $\mathcal{C}$ ones:$\mathcal{M} \subseteq \mathcal{C}$, a new schema of maximal connected frequent unique labelled subgraphs is proposed. We investigate in the problem of listing the family of all support-closed connected frequent subgraphs to find the maximal $\mathcal{M}$ ones of them. This choice is motived by the fact that: first, the generation of a connected unique labelled (n+1) edges from one of its proper (n) edges subgraph requires a polynomial time, second, listing the maximals $\mathcal{M}$ by just visiting the closed ones $\mathcal{C}$ will be well optimised to visit all the frequents ones $\mathcal{F}$. Listing the family of all support-closed patterns of a non-redundant dataset $\mathcal{D}$ of subgraphs of a connected graph G has been studied by Boley (2011), but this result can not applied in the case where a database is a set of connected graphs, this is due to the failure subgraph isomorphism. To overcome this later phenomena we reuse the canonical encoding of unique labelled subgraphs proposed by Thomas et al. (2009). This later encoding reduce the problem of subgraph isomorphism to a subset relation test for which just a polynomial time is required. As consequence it can be trivially seen that the proposed approach generates subgraphs and calculates their closure in polynomial time.

Besides this introduction, this paper is organized as follows: the following section presents the formalism used in the article. In Section 3; an overview of closed and maximal connected frequent subgraphs mining algorithms is represented. In Section 4; subgraphs system which represents connected subgraphs of a graph G is studied, and we present our mining schema. In Section 5, We reach our conclusion.

2 Preliminary

To understand the previous proposition, some basic background knowledge is required.

2.1 Graphs notations

Definition 1. ***(Graph)*** *A graph $G = (V, E)$ consists of V a vertices set and an edges set $E \subseteq V \times V$ such that each edge $e \in E$ is associated with an unordered pair of vertices.*

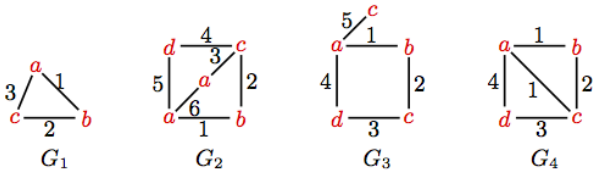


FIG. 1 – *Undirected labelled connected graphs*

We defined a labelled graph by added a labels set Ψ and a function $\Gamma : V \cup E \rightarrow \Psi$ assigned for each vertex and edge a label of Ψ.

Definition 2. ***(Labelled graph)*** *A graph $G = (V, E, \Psi, \Gamma)$ is a labelled graph where V a vertices set, $E \subseteq V \times V$ the edges set, Ψ a labels set and Γ is a function assigning labels to all edges and nodes.*

The graph is called undirected graph, i.e. the edge (u_1, u_2) is equal to (u_2, u_1). An edge e (u_1, u_2) has two end-points u_1, and u_2, and the edge e is incident to u_1 and u_2. Two nodes are adjacent if they share the same end-points. The degree of a vertex is the number of incident edges on the vertex v and is denoted by $deg_G(v)$. A vertex of degree one, is called an end vertex. A path is an edges sequence $\{(v_1, v_2), ..., (v_{n-1}, v_n)\}$ if there is an edge v_x between v_n and v_1 the path is a cycle. The graph is connected if there is a path between any two vertices of V i.e. there is no vertex $v_i \in V$ where $deg_G(v_i)$=0.

Definition 3. ***(Unique edge labelled graph)*** *A graph $G = (V, E, \Psi, \Gamma)$ is an unique edge labelled if and only if each edge label occurs at most once.*

Example 1. *As instance the figure 1 shows the unique edge labelled graphs G_1, G_2, G_3, but the graph G_4 is not a unique edge labelled graph because the two edges $a - b$ and $a - c$ have a same label* 1.

Definition 4. ***(Isomorphism)*** *A graph isomorphism from $G_1 = (E_1, V_1)$ to $G_2 = (E_2, V_2)$ is a bijection function φ from E_1 to E_2 in which : $\forall (i, j) \in V_1 \Longleftrightarrow (\varphi(i), \varphi(j)) \in E_2$.*

Definition 5. ***(Subgraph isomorphism)*** *$G_1 = (E_1, V_1)$ is a subgraph isomorphism into $G_2 = (E_2, V_2)$ if there is a bijection function φ from G_1 to $G'_2 \subset G_2$.*

2.2 Closure operator

Let $(E, \mathcal{H})$ be a set system. A mapping $\sigma : \mathcal{H} \rightarrow \mathcal{H}$ is called a closure operator if it satisfies for all $X, Y \in \mathcal{H}$ that

- $X \subseteq \sigma(X)$(extensivity);
- $X \subseteq Y \rightarrow \sigma(X) \subseteq \sigma(Y)$ (monotonicity);
- $\sigma(X) = \sigma(\sigma(X))$(idempotence).

In data mining i.e (pattern mining) the closedness is defined as support-closed patterns of a dataset. Given a transactional database, a pattern is closed in a data set if there exists no super pattern that has the same support count as this original pattern.

2.3 Set System Properties

In this section we define the notions and notations of set systems proprieties used in this paper.

Definition 6. *(Set system). A set system is an ordered pair $(E, \mathcal{H})$, where E is the ground set and $\mathcal{H}$ a no empty subset of the power set of some finite set E, $\mathcal{H} \subseteq 2^E$. (non-empty) Set system $(E, \mathcal{H})$ is called:*

- **a closed system** if $E \in \mathcal{H}$, and $X, Y \in \mathcal{H}$ implies $X \cap Y \in \mathcal{H}$;
- **an accessible** if for all $X \in \mathcal{H} \backslash \{\emptyset\}$ there is an $e \in X$ such that $X \backslash \{e\} \in \mathcal{H}$;
- **a strongly accessible** if for every $X, Y \in \mathcal{H}$ satisfying $X \subset Y$, there is an $e \in Y \backslash X$ such that $X \cup \{e\} \in \mathcal{H}$;
- **an independence** if $Y \in \mathcal{H}$ and $\forall X \subseteq Y \rightarrow X \in \mathcal{H}$;
- **a greedoid** if it is accessible and satisfies the augmentation property, i.e., for all $X, Y \in \mathcal{H}$ with $|X| < |Y|$, there is an element $e \in Y \backslash X$ such that $X \cup \{e\} \in \mathcal{H}$;
- **a matroid** if it is a greedoid and an independence system;
- **a confluent** if $\forall I, X, Y \in F$ with $\emptyset \neq I \subseteq X$ and $I \subseteq Y$ it holds that$X \cup Y \in \mathcal{H}$.

3 Maximal and closed connected subgraphs mining algorithms

In the existing literature, there are not many algorithms developed to face the exponential number of connected frequent subgraphs that formed the search space for finding closed and maximal frequent subgraphs. In the remaining of this paper we use in short graph to indicate connected graph. A typical approach in order to mine the closed and maximal frequent subgraphs is by finding all frequent sub-graphs incrementally in an Apriori or DFS manner to get the closed and maximal frequent ones, but this process is not efficient because $|\mathcal{F} \cup Bd^{-}(\mathcal{M})|$ subgraphs need to be visited in the aim to find the maximal frequent ones. In order to avoid the exploration of all those frequent subgraphs the existing closed and maximal frequent subgraphs mining algorithms use several pruning techniques to reduce the search space (subgraphs system). In this section, we discuss four of the most popular algorithms that were proposed to solve this discussed problem: CloseGraph (mining the closed frequent subgraphs); SPIN (SPanning tree based maximal graph mINing),MARGIN (mining maximal frequent subgraphs), and ISG (Mining maximal frequent graphs using itemsets).

3.1 CloseGraph

In the best of our knowledge the famous algorithm to find the closed sub-graphs is CloseGraph Yan and Han (2003), this algorithm behaves recursively using the depth- first search and right-most extension like gSpan Yan and Han (2002), with the addition of a graph pruning technique in order to avoid the generation of all frequent graphs. This pruning technique is based on the early termination condition that if it is satisfied, then the searching process will be completely stopped for some descendant branches, which effectively reduce the search space and enumerate only closed subgraphs. The early termination condition is based on the equivalence occurrence propriety which permeates to CloseGraph to decide if the descendants super-graphs of a graph X will not be closed. This early termination condition does not always hold. The exploration process will miss some frequent subgraphs. This phenomenon is

noted as failure of early termination, where the experimentation results have illustrated that the detection of the failure cases requires a half of the performance.

3.1.1 CloseGraph newness

As we see, gSpan adopts a depth-first based search for finding all frequent subgraphs, and CloseGraph is built on gSpan but it is interested by only the closed frequent subgraphs. In order to avoid the process of finding the complete frequent graph set first, and then filter it to obtain the closed ones, GloseGraph proposes a trade-off between gSpan search technique and the early termination filter condition. This filter condition prune the search space in the target to mine the closed frequent subgraphs as directly as possible, by minimizing visiting DFS codes of frequents subgraphs g that are not closed. This minimization is got by terminating the extension of a subgraph g to its supergraph g' by pruning all ascendants of the first one iff g and g' have an equivalence occurrence. The equivalent occurrence propriety means wherever a subgraph g occurs in $\mathcal{D}$, its supergraph immediate g' occurs, then we need not grow from a subgraph g but from its supergraph g'. The early termination does not hold for every supergraph. Those supergraphs are noted failure points of early termination condition which need to be detected to guarantee the completeness of mining results. This later setp requires much computational cost.

3.1.2 CloseGraph discussion

CloseGraph can be mainly summarized to four tasks which play a major role in the total cost of the algorithm: (1) Check if the DFS code is minimal ($DFS(s) = DFS_{min}(s)$), (2) Subgraph isomorphism test queries , (3) Check the early termination condition and its failure cases, (4) Extending every graph g candidate in any possible position to generate the supergarphs set first, and then filter it to obtain only which are a right-most extended from g.

Using gSpan DFS coding reduce the graphs mining problem to mining their corresponding DFS_{min} codes, which turns to be a sequential patterns mining problem, but the exponential number of possible duplicates implies that in the worst case, CloseGraph requires an exponential number of the costly $DFS(s) = DFS_{min}(s)$ test i.e (task (1)) which is unnecessary computation on duplicates. Subgraph isomorphism is another factor that increases the total computational cost of CloseGraph. The subgraph isomorphism is an NP-complete problem, and there is no polynomial algorithm to solve it. Since in the best case CloseGraph requires $\mathcal{O}(|E|^2\ |\mathcal{D}|)$ subgraph isomorphism test queries for every subgraph visited, w.r.t. early termination condition for which the equivalence occurrence is tested by checking the occurrence of the subgraph and its supergraphs i.e. (task 2 and the first part of the task 3). The attempting process to detect the failure cases of the early termination largely increases the computational cost. This process requires a trade-off between the great amount of information needs to be stored and the large number of subgraphs frequency calculates for all subgraphs that have been visited, because the detection technique used by CloseGraph works in a passive way (the second part of the task 3).

3.2 SPIN

For the case of finding only the maximal frequent subgraphs there are two references algorithms SPIN Huan et al. (2004) and Margin Thomas et al. (2010). Spin, first all frequent trees are mined from a graph database and then the groups of frequent subgraphs are constructed from those mined trees. Every group of frequent subgraphs forms an equivalence class. Every equivalence class is composed by subgraphs which share the same canonical spanning tree. However, it is easy to see that, this latter step is not efficient because the enumeration of all frequent subgraphs is still required in order to construct maximal frequent ones , to avoid this problem some optimization techniques (Buttom-Up Pruning, Tail Srink, and External-Edge Pruning) are integrate to speed up the mining process.

3.2.1 SPIN discussion

After this illustration of SPIN, we now discuss its operational behavior. Generally, in brief summary, SPIN complexity depends mainly on the cost of: (1) all frequent subtrees mining, (2) a set of frequent subgraphs is reconstructed, and maximal local subgraphs are mined (Maximal in an equivalence class) by including optimization techniques, and (3) Maximal global subgraphs discovery.

The efficiency of maximal frequent patterns mining algorithms is deduced according to their performance to reduce subgraphs visited below the border, and the reduction cost must be reasonable. SPIN advocate all frequent trees $|\mathcal{T}_{\mathcal{F}}|$ mining first, because tree related operations are simpler than graphs ones i.e.(isomorphism , canonisation). Generally subtree isomorphism is polynomial in contrast to subgraph isomorphism which is NP-complete. However, despite this motivation, experiments on many databases (sparse graphs data and high support threshold) have been illustrated that most of frequent subpatterns are really trees Nijssen and Kok (2005)Thomas et al. (2010), the number of visited subtrees is exponential which implies an exponential number of subtree isomorphism queries. The complexity of this first task of SPIN by using an algorithm which use only is-frequent queries i.e(we ignore the generation task) is $\mathcal{O}(|\mathcal{T}_{\mathcal{F}} + \mathcal{M}(\mathcal{T}_{\mathcal{F}})|\ |\mathcal{D}|\ T_{sti})$, where T_{sti} is the time cost of subtree isomorphism test.

For each subtree $t \in \mathcal{T}_{\mathcal{F}}$, SPIN reconsecrates a set of subgraphs that share t as canonical spanning tree i.e (equivalence class). As presented above the equivalence class is a lattice search space i.e (powerset) of candidate edges C. Traversing all subsets in the search space necessities $\mathcal{O}(|2^{|C|}\ (T_g + |\mathcal{D}|\ T_{sgi})|)$, where $T_g,\ T_{sgi}$ are the generation of joined graph $t \oplus c$ and subgraph isomorphism test querier times respectively. To overcome this problem SPIN integrates several optimization techniques which can prune the entire equivalence class. Those techniques offer asymptotic computational cost in order to find maximal local frequent subgraphs. First, SPIN optimizes the number of equivalence classes explored by using External-Edge Pruning, where each tree $t \in \mathcal{T}_{\mathcal{F}}$ is not explored if it contains an external edge. This operation is required for every tree $t \in \mathcal{T}_{\mathcal{F}}$ which implies $\mathcal{O}(|\mathcal{T}_{\mathcal{F}}|\ |V|\ |\mathcal{D}|\ T_{vc})$ where T_{vc} is the necessary time for checking the existence of an external edge connected a node $v \in V \in t$ and a node which is not in t.

After this first filtering process, for each subtree t of the retained subtrees, candidate edges set C will be calculated which requires $\mathcal{O}(|E|\ (T_g + |\mathcal{D}|\ T_{sgi}))$. Then, from this set C a set of associative edges A is seek, for which wherever t occurs in $\mathcal{D}$, $t \oplus A$ must also occur exactly in the same place. Thus mean t must be accompanied with A in any place where t occurs.

As conclusion every supergraph of t must contain A which reduces the research space, where the complexity of finding A is $\mathcal{O}(|C|\ (T_g + |\mathcal{D}|\ T_{sgi}))$. As exposed in this section if a set of associative edges A is *lethal* then the entire equivalence class reconstructed from t will be pruned, which means to not explore $t \oplus A$ research space. Checking if A is a *lethal* implies the calculation of the canonical spanning tree of the graph $t \oplus C$.

Next to this latter optimization technique, SPIN invokes *Bottom-Up Pruning* technique in order to speed up the exploration of the equivalence class of t, in the target to find the maximal local frequent subgraphs $\mathcal{LM}$ as quickly as possible. This technique necessities to check the frequent largest possible subgraph $g = t \oplus c$, where in the best case requires $\mathcal{O}(T_g + |\mathcal{D}|\ T_{sgi})$ where $c = C$ which implies that there is an unique maximal frequent subgraph of the equivalence class of t. Finally, SPIN finds the global maximal set by removing all subgraphs that are not maximal due to their are a proper subgraph of another local maximal frequent subgraph, using this final step, SPIN outputs the global maximal frequents subgraphs set in $\mathcal{O}(|\mathcal{LM}|^2\ |T_{sgi}|)$. All of these processing explains the high computational complexity of SPIN in the target to find all maximal frequent subgraphs.

3.3 MARGIN

Margin is another graph mining algorithm that mines only maximal frequent subgraphs, motivate by the idea that generally the maximal frequent subgraphs lie in the middle of the research space (subgarphs system), which implies highly computational cost if a typical approach is used in order to find maximal frequents subgraphs, which would require bottom-up traversal of the research space wherein the frequent subgraphs that are not maximal. To overcome this problem, Margin avoids the exploration of the space above or below border (Maximal frequent and minimal non frequent subgraphs) and visits only the frequent subgraphs (promising subgraphs $\widehat{\mathcal{F}}$) that lie on the border of the frequent and infrequent subgraphs. Those frequent subgraphs noted promising subgraphs that lie on the border are the set of n-$edge$ frequent subgraphs that have an $n+1$-$edge$ infrequent supergraph.Technically, Margin seeks to locate one promising subgraph, and then jump from one promising to another, until exploring all the promising subgraphs of the subgraphs system, finally all maximally frequent subgraphs are finded by retaining from those promising subgraphs only the subgraphs that have all their immediate supergraphs are infrequent.

In brief summery, for each graph $G_i \in \mathcal{D}$ Margin is a framework which process as follows: first the graph system (research space) is explored in a depth first in order to find the representative R_i, and then invoking ExpandCut method, initially on the cut CR_i and R_i where CR_i is a supergraph infrequent of R_i. ExpandCut finds the nearby cuts and recursively calls itself on each newly found cut, until any new cut can be found. Given a family of cuts $(C|P)$ in that ExpandCut is invoked, the frequent subgraphs P for each cut are reported as promising frequent subgraphs $\widehat{\mathcal{F}}$, and then only maximal local frequent subgraphs $\mathcal{ML}$ are retained from the promising $\widehat{\mathcal{F}}$ ones. Finally, the maximal global frequent subgraphs $\mathcal{M}$ in the data base $\mathcal{D}$ are found by removing the set of graphs in $\mathcal{ML}$ which are proper subgraph of another frequent graph in $\mathcal{ML}$.

3.3.1 MARGIN discussion

After this illustration of MARGIN we now discuss its computational cost. In brief summary, MARGIN complexity depends mainly on the cost of: (1) finding the first promising subgraph, (2) invoking *ExpandCut* for a new nearly cut , (3) Selecting maximal locally frequent subgraphs $\mathcal{ML}$, and (4) Maximal global subgraphs discovery.

For the first step, MARGIN seeks to find the representative R_i for each $G_i \in \mathcal{D}$. This is done by iteratively dropping or joining an edge from G_i or from $\emptyset$ respectively, until a frequent subgraph, or an infrequent subgraph is found. In those two cases, finding the representative R_i can be done on $\mathcal{O}(|E|\ (T_g + |\mathcal{D}|\ T_{sgi}))$, where $T_g,\ T_{sgi}$ are the generation and subgraph isomorphism test querier times respectively.

The second principal task of MARGIN is finding all promising subgraphs $\widehat{\mathcal{F}}$ for each graph $G_i \in \mathcal{D}$. In order to find the set of promising subgraphs, *ExpandCut* is invoked, where for each invocation a promising subgraph $\widehat{f}$ is reported. Generally, two successive invocation of *ExpandCut* can be done in a delay of $\mathcal{O}(|E|\ (T_g + |\mathcal{D}|\ T_{sgi} + |E|))$ in the worst case. From this delay we can deduce that MARGIN can outputted the set of promising subgraphs $\widehat{\mathcal{F}}$ of an graph $G_i \in \mathcal{D}$ in $\mathcal{O}(|E|\ (T_g + |\mathcal{D}|\ T_{sgi} + |E|)\ |\widehat{\mathcal{F}}|)$.

From those promising subgraphs $\widehat{\mathcal{F}}$, Margin selects only the maximal local ones $\mathcal{ML}_i$ for every graph $G_i \in \mathcal{D}$ by removing subgraphs that are contained in at least another promising subgraph. This latter step can be done by a naive algorithm that test all possibility in $\mathcal{O}(|\widehat{\mathcal{F}}|^2\ T_{sgi})$. Finally, the complete maximal local frequent subgraphs $\mathcal{CML}$ set ($\cup\mathcal{ML}_i$ for all $G_i \in \mathcal{D}$) will be filtered to get just the set of the maximal globally frequent subgraphs $\mathcal{M}$, which require in the worst case $\mathcal{O}(|\mathcal{CML}|^2\ T_{sgi}\ |\mathcal{D}|)$.

3.4 ISG

ISG is an algorithm for mining maximal frequent subgraphs over a database of graphs having unique edge labels Thomas et al. (2009). In order to find those maximal frequent subgraphs, itemset mining technique is used. Mainly the idea is to transform the problem of maximal graphs mining to maximal itemset mining. First, the graphs database is transformed into transaction sets data, where each graph in $\mathcal{D}$ is encoded as a set of items. These items represent the edges and the converse edges of the graph and 3-edges substructure that are contained in the graph i.e.(triangle, spike, and linear chain). Aptness these 3-edges blocks are called secondary structures and are added to avoid the problem of edge triplets and converse edge triplets conversion to graph, the edge triplets and converse edge triplets together do not guarantee that the given maximal frequent itemset be converted into a graph unambiguously. Each secondary structure is assigned to a unique item id in addition to the unique ids of the edge and the converse edge triplets that compose the 3-edges block.

Example 2. *As instance the code of the graph G_1 illustrated in the figure1, $code(G_1) = \{(a,1,b),(b,2,c),(b,3,c),(1,b,2),(2,b,3),(1,a,3),trip1,tringle)\}$, where $trip1$ is the common item identifier of the edge and the converse edge triplet $trip1 = \{(a,1,b),(b,2,c),(b,3,c),(1,b,2),(2,b,3),(1,a,3)\}$, and the item $triangle$ signifies that the 3-edges block is a triangle.*

Second, the transaction database $\mathcal{D}'$ is used as input of an maximal itemsets mining algorithm, the set of the maximals code itemsets $\mathcal{M}_I$ is enumerated, but generally there is any

guaranty that all these codes correspondent to a unique connected subgraph i.e.(there is no bijection between $\mathcal{M}_I$ and an subset of subgraphs in $\mathcal{H}$). The principal problem over those $\mathcal{M}_I$ is that there is codes for which the conversion generate disconnected graphs and as mentioned earlier the problem of the conflicting maximal frequent itemset for which the conversion of one code can result many subgraph candidates. To avoid those problems an preprocessing step after the conversion of the codes that can not be converted unambiguously is required. This later step consist to converse a disconnected code to a set of its connected graphs, and for the case of conflicting maximal frequent itemset, it is breaked to form non-conflicting subsets. In continuation of this conversion step, the set of the subgraphs generated is filtrated to prune the subgraphs that are contained in other subgraphs. Finally, the set of maximal connected unique labelled edge subgraphs.

3.4.1 ISG discussion

After this illustration of ISG we now discuss its computational cost. In brief summary, ISG complexity depends mainly on the cost of: (1) encoding the graphs database $\mathcal{D}$ into transactional data $\mathcal{D}'$, (2) Maximal itemsets mining over $\mathcal{D}'$, (3) converting the maximal itemsets to connected subgraphs, and (4) pruning no maximal subgraphs.

For the first step, it is trivial to see that the encoding task requires $\mathcal{O}((|E|\ |V| + |E|^3)\ \mathcal{T}_i)$, where T_i is the item affectation time. The second principal task of ISG can be achieved by using any algorithm of maximal itemsets mining where in the best case $\mathcal{O}(|\mathcal{M}_I \cup bd^-(\mathcal{M}_I)|\ \mathcal{T}_{sr})$ where $\mathcal{T}_{sr}$ is the subset relation test required time. But, in the worst case this latter step required $\mathcal{O}(|\mathcal{F} \cup bd^-(\mathcal{M}_I)|\ \mathcal{T}_{sr})$.

The conversion step play a major role in order to achieve the maximal connected frequent unique labelled subgraphs, all edges and converse edges of the graph code are visited where the corresponded connected graph is generated or the set of the largest connected disjoint components of the code are generated for a disconnected graph code. For this latter step $\mathcal{O}(|E|^2\ \mathcal{T}_i)$ is required for each maximal itemset code. From these generated connected subgraphs $\mathcal{CM}$ the maximal ones are outputted in $\mathcal{O}(|\mathcal{CM}|^2\ \mathcal{T}_{sr})$.

4 Maximals unique labelled subgraphs mining

After this illustration of the existing algorithms used to mine the maximal and the closed connected subgraphs, In this section we present the proposed mining schema. Mainly our idea is to enumerate the maximal connected frequent unique labelled subgraphs by visiting just the closed subgraphs $\mathcal{C}$ and by avoiding the use of the subgraph isomorphism test. In the aim to get the maximal subgraphs the proposed idea can be summarized as: first a preprocessing phase is required where each graph in $\mathcal{D}$ is traversing to encode it as a set of edge and converse edges codes i.e.(transactional dataset $\mathcal{D}'$). Next, according to the strongly accessibility propriety, divide-and-conquer algorithm is used to enumerate the closed frequent connected subgraphs by vesting only the inductive generator subgraphs. Then each visited inductive generator subgraph g' will be encoded as edge and converse edge code ($code(g')$) which will be used to test frequency, then computing the closed successive code ($code(g'')$) of $code(g')$ which is equivalent to $\cap\mathcal{D}[code(g')]$. After computing $code(g'')$, it will be decoded to the maximal connected subgraph g'' contained g'. These steps will be repeated until there is not any more unvisited

inductive generator subgraph. Finally, the set of the closed frequent subgraphs is merged in order to reach the maximal ones over them i.e. $\mathcal{M} = \{\forall c \in \mathcal{C}, \nexists c' \in \mathcal{C} | c \subset c'\}$.

4.1 Connected Subgraphs System

As mentioned earlier the search space of a given problem play a key role to decide if the efficient algorithms developed for discovering maximal and closed frequent patterns where search space is a lattice i.g, (dualize and advance) or an independence set system or at least strongly accessible i.g, (Divide & Conquer) can be applied to this problem. In this subsection, we turn to the complexity of unique edge label subgraphs research space system, where the strongly accessibility of the subgraphs system is proofed.

Theorem 1. *Unique edge label subgraphs system is not closed and is not a lattice.*

Proof. There is no assurance that the intersection of two connected subgraphs results a connected subgraph. □

By using this theorem we can deduce that we can not apply the efficient enumeration algorithm dualize and advance which requires that the underlying pattern system is a lattice or a chain product. According to this conclusion, and that the maximal frequents connected subgraphs $\mathcal{M}$ are contained in the set of the closed $\mathcal{C}$ ones, we attempt to verifier if Divide & Conquer can be applied in order to find the closed ones which will be filtered to find the maximal ones only. This choice is motived by the number of closed frequents patterns is generally largely smaller than the number of the frequents ones $\mathcal{F}$. In this attempt to find the closed ones first, then the maximal ones, the strongly accessibility propriety of the patterns set system is required. A set system is strongly accessible this means that every $X \in \mathcal{H}$ can be reached from all $Y \subset X$ with $Y \in \mathcal{H}$ via augmentations with single elements inside $\mathcal{H}$.

Theorem 2. *Unique edge label subgraphs system $\mathcal{H}$ is strongly accessible.*

Proof. An unique edge label subgraphs system $\mathcal{H}$ is accessible, let X a connected subgraph where $X \in \mathcal{H}$, if X is a graph that contains a cycle then we just drop an edge e from the cycle and the result $X \backslash e \in \mathcal{H}$. Otherwise, if X is without a cycle (a tree), we just drop one of its leaves.
Given two graph $X_1, X_2 \in \mathcal{H}$ with $X_2 \subset X_1$. Assume that there exist not an edge $e \in X_1 \backslash X_2$ such as $X_2 \cup e \in \mathcal{H}$. So X_2 and $X_1 \backslash X_2$ are two components disconnected in $G[X_1]$ which contradicting the choice of X_2. □

So, this propriety improve the enumeration process because any closed frequent pattern can be reached from any one that have been already found by one element augmentation Boley (2011)Boley et al. (2007), where the follow theorem is given:

Theorem 3. *"for any finite strongly accessible set system $(E, \mathcal{H})$ given by a polynomial membership oracle and for any polynomially computable closure operator $p : \mathcal{H} \to \mathcal{H}$, the family $p(\mathcal{H})$ of p-closed sets can be enumerated with polynomial delay."*

According to this theorem, it can be trivially seen that the closure operator is not polynomially computable. So, the enumeration delay is not polynomial if the subgraph isomorphism test is used.

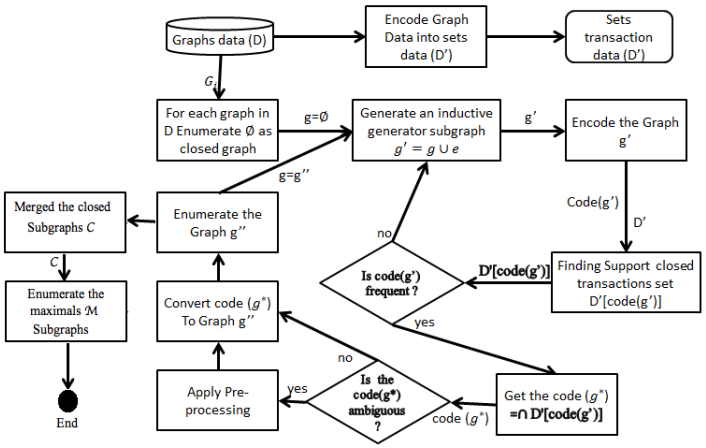


FIG. 2 – *Maximals unique labelled subgraphs mining schema*

4.2 The maximals unique labelled subgraphs mining approach

In this subsection our proposed schema for maximal connected frequent unique labelled edge subgraphs mining is presented. Figure 2 shows an overview of the proposed approach, which can be broadly divided into four principals steps. Step 1 converts each graph in the database D into an edges and converse edges set transaction. This conversion step involves mapping parts of the graph to edges and converse edges,any edge e is represented as a 3-tuple $(et_{v_i}, et_e, et_{v_j})$, where et_{v_i}, et_{v_j} are the two labels of the two vertices connected by e, (v_i, v_j), and the label of e respectively. Clearly the conversion of any graph g into a transaction t is an important task, but related to the objective we should be able to reconstruct g using t which is not guarantee as illustrated in Thomas et al. (2009). There is a need for mapping of additional substructures of the graph in order to ensure unambiguous conversion of the transaction codes into graphs. These additional substructures are called secondary structures and can be presented as: first, an unique item assigned to each three connected edges, where each isomorph three edges and converse edges code will have the same common id item. Second, a type item code to precise the type of the three edge code i.e.(triangle, spike, linear chain), for more detail see Thomas et al. (2009).

In step 2, as mentioned earlier, the strongly accessibility of the subgraphs system is investigated in order to apply the divide-and-conquer algorithm based on the notion of inductive generator subgraph in order to list the closed support connected subgraphs $\mathcal{C}$ by visiting only $|\mathcal{C}|$, which largely optimise the number of frequency test query. So, in this step first, starting by the $\emptyset$ subgraph, for each closed support listed above $g \in \mathcal{C}$ we attempt the generation of a new connected subgraph g' i.e.(inductive generator), this later task is reached by adding a new edge e i.e.($g' = g \cup e$). Then, the generated g' subgraph is converted to an edges and converse edges code i.e.($code(g')$ with secondary structures). Next, $code(g')$ is test over the converted data $\mathcal{D}'$ by considering the subset relation which is polynomial, compare to subgraph isomorphism relation which is NP-complete. Following this latter task the largest code i.e.($code(g'')$) including $code(g')$ and has not decreased the support according to the anti-monotonicity of the

frequency, $\sigma(code(g')) = code(g'') = \cap\mathcal{D}'[code(g')]$ which is polynomail in the size of $\mathcal{D}'$. After reaching the code $\sigma(code(g')) = \cap\mathcal{D}'[code(g')]$, converting this code to an connected subgraph is not guarantee, this is due to: (1) the code does not contain the type item code, and/or (2) the code is converted to an disconnected subgraph.

In the step 3 the two cases of ambiguity are handled, where for the first case an preprocessing is invoked. The preprocessing phase breaks the conflicting code to form non-conflicting subsets to hold the larger connected subgraphs that contains $code(g')$. For the second case the $code(g'')$ is converted to an disconnected graph but just the largest connected component that includes $g' = g \cup e$ is generating. The step 2,and 3 are repeated if there is any more connected unvisited subgraph g''.

In the step 4, besides this presentation of the steps follow in the target to enumerate the closed frequent connected subgraphs $\mathcal{C}$, this later set of subgraph is explored where the subgraph contained in another subgraph in $\mathcal{C}$ are eliminated. Finally just the set of the maximal connected subgraphs $\mathcal{M}$ is outputted. This later step cost at most $\mathcal{O}(|\mathcal{C}|^2\,\mathcal{T}_{sr})$ where $\mathcal{T}_{sr}$ is the time required to test subset relation.

5 Conclusion

In this paper, we have presented a complexity results study of connected closed and maximal frequent subgraphs mining in literature. In this context, we have illustrated that the pattern search space system and the occurrence frequency checking complexity decrease the enumeration process robustness of the closed and maximal frequent patterns mining algorithms.

On the first hand, The complexity of the checking frequency subgraph isomorphism reduces the efficiently of the enumeration of frequency subgraphs compare to subset relation where closed and maximal frequent subsets are seek. On the second hand, the graph search space system do not check some proprieties used to speed up the enumeration process. Connected subgraphs system is not a lattice nor a chain product. Following this idea the dualization technique can not be apply to seek connected maximal frequent subgraphs, in contrast to the set system where dualize and advance algorithm have the best efficiently in the aim to find the maximal frequent subsets.

Finally we have shown that the unique edge label connected subgraphs system of a given graph is strongly accessible. We have investigated this propriety to reduce the number of visited subgraphs before the maximal ones to $|\mathcal{C}|$. The subgraph isomorphism problem was avoided, where encoding the graphs as edges and converse edges set is reused. This encoding reduce the problem of subgraph isomorphism to a subset test. We have deduce that finding closed unique edge label connected subgraphs, then select the maximal over them is more efficiency then finding maximal subgarphs by using the existing algorithms.

References

Agrawal, R., T. Imieliński, and A. Swami (1993). Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*, Volume 22, pp. 207–216. ACM.

Boley, M. (2011). *The Efficient Discovery of Interesting Closed Pattern Collections.* Ph. D. thesis, University of Bonn.

Boley, M., T. Horváth, A. Poigné, and S. Wrobel (2007). Efficient closed pattern mining in strongly accessible set systems. In *Knowledge Discovery in Databases: PKDD 2007*, pp. 382–389. Springer.

Chi, Y., R. R. Muntz, S. Nijssen, and J. N. Kok (2005). Frequent subtree mining-an overview. *Fundamenta Informaticae 66*(1), 161–198.

Ganter, B. and K. Reuter (1991). Finding all closed sets: A general approach. *Order 8*(3), 283–290.

Gunopulos, D., R. Khardon, H. Mannila, S. Saluja, H. Toivonen, and R. S. Sharma (2003). Discovering all most specific sentences. *ACM Transactions on Database Systems (TODS) 28*(2), 140–174.

Gunopulos, D., H. Mannila, R. Khardon, and H. Toivonen (1997). Data mining, hypergraph transversals, and machine learning. In *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pp. 209–216. ACM.

Huan, J., W. Wang, J. Prins, and J. Yang (2004). Spin: mining maximal frequent subgraphs from graph databases. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 581–586. ACM.

Nijssen, S. and J. N. Kok (2005). The gaston tool for frequent subgraph mining. *Electronic Notes in Theoretical Computer Science 127*(1), 77–87.

Thomas, L., S. Valluri, and K. Karlapalem (2009). Isg: Itemset based subgraph mining. Technical report, Technical Report, IIIT, Hyderabad, December2009.

Thomas, L. T., S. R. Valluri, and K. Karlapalem (2010). Margin: Maximal frequent subgraph mining. *ACM Transactions on Knowledge Discovery from Data (TKDD) 4*(3), 10.

Yan, X. and J. Han (2002). gspan: Graph-based substructure pattern mining. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pp. 721–724. IEEE.

Yan, X. and J. Han (2003). Closegraph: mining closed frequent graph patterns. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 286–295. ACM.

Résumé

Le problème de découverte des motifs maximaux fréquents peut être résolu dans un délai polynomial pour les ensembles, les sequences, les arbres, etc. Par ailleurs, pour les sous graphes connexes fréquents le problème devient intraitable. Les approches existantes ont proposé des différentes techniques afin de réduire l'espace de recherche et le nombre de motifs à visiter pour atteindre les maximaux, ce qui réduira par la suite le nombre de tests de sous graphes isomorphes. Nous présentons dans cet article une étude des algorithmes de découverte de sous-graphes maximaux connexes fréquents, et une nouvelle approche pour la découvertes des sous graphes maximaux connexes fréquents dans une base de graphes a étiquetage d'arrêtes unique. La finalité étant de réduire le nombre des sous graphes visités avant l'extraction des maximaux et de ramener le test exponentiel d'isomorphisme de sous graphes à un test de sous ensembles d'un ensemble qui lui est polynomial.

Link prediction in multiplex networks: application to co-authorship link prediction in bibliographical networks

Manisha Pujari*

* LIPN CNRS UMR-7030,
Université Sorborne Paris Cité,
manisha.pujari@lipn.univ-paris13.fr

Résumé. Nous nous intéressons dans ce travail, au problème de la prévision de liens dans les graphes de co-publication ou de collaborations scientifiques. A la différence des approches existantes fondées sur l'analyse de seuls graphes de co-publicaction, nous proposons dans ce travail, d'étendre l'analyse à d'autres types de relations reliant les auteurs, en particulier la relation de co-participation à une même conférence et la relation de co-citation de mêmes références bibliographiques. Ainsi, nous considérons ici un réseau multiplex formé de trois couches (une couche par type de relation considérée), et nous montrons que l'extension des approches dyadiques classiques pour la prévision de liens à ce type de réseaux multiplexes peut améliorer les résultats de prévision de collaboration scientifique.

1 Introduction

Link prediction plays an important role in the analysis of complex networks with its wide range of application like identification of missing links in biological networks, identification of hidden and probable new criminal links, recommendation systems, prediction of future collaborations or purchases in e-commerce. It can be defined as the process of identifying missing or new links in a network by studying the history of the network. A popular category of link prediction approaches is *dyadic topological approaches* which consider only the graph structure and computing a score for unconnected nodes pairs, they predict the possibility of getting new links. In a seminal work proposed in (Liben-Nowell et Kleinberg, 2007), authors have shown that simple topological features characterizing pairs of unlinked nodes, can be used for predicting formation of new links. They propose to sort a list of unconnected node pairs according to the values of a topological measure. The top k node pairs are then returned as the output of the prediction task. Here an assumption is made that the topological measure should be able to rank the most probable new links on the top. Many other works have been published focusing on combining different topological metrics to enhance prediction performances which convert the link prediction task to a binary classification problem and hence use machine learning (Hasan et al., 2006; Benchettara et al., 2010).

But all these work address the link prediction in only simple networks which have homogeneous links. To our knowledge, not much have been explored to add multiplex information

for the task of link prediction. Although there are few recent works that propose methods for prediction of links in heterogeneous networks, networks which have different types of nodes as well as edges (Yizhou Sun et al., 2011). There have also been few work on extending simple structural features like degree, path etc. to the context of multiplex networks (Battiston et al., 2013; Berlingerio et al., 2011) but none have attempted to use them for link prediction. We propose a new approach for exploring the multiplex relations to predict future collaboration (co-authorship links) among authors. The applied approach is a supervised-machine learning approach where we attempt to learn a model for link formation based on a set of topological attributes describing both positive and negative examples. While such an approach has been successfully applied in the context on simple networks, different options can be applied to extend it to the multiplex network context. One option is to compute topological attributes in each layer of the multiplex. Another one is to compute directly new multiplex-based attributes quantifying the multiplex nature of dyads (potential links).

2 Link prediction approach

Our approach includes computing simple topological scores for unconnected node pairs in a graph. Then we extend these attributes to include information from other dimension graphs. This can be done in three ways : First we compute the simple topological measures in all dimensions ; Second is to take the average of the scores ; and Third we propose an entropy based version of each topological measures which gives importance to the presence of a non-zero score of the node pair in each dimension. In the end all these categories of attributes can be combined in various ways to form vectors of attribute values characterizing each example or unconnected node pair. Formally, if we have a multiplex graph $G =< V, E_1, \ldots, E_m >$ which in fact is a set of graphs $< G_1, G_2, \ldots, G_m >$ and a topological attribute X. For any two unconnected nodes u and v in graph G_i (where we want to make a prediction), $X(u, v)$ computed on G_i will be *direct* attribute and the same computed on all other dimension graphs will be *indirect* attributes. The second category computes an average of the attribute over all the dimension i.e. $Average(X) = \frac{\sum_{\alpha=1}^{m} X(u,v)^{[\alpha]}}{m}$ for $u, v \in V$ and $(u, v) \notin E_i$. where m is the number of different types relation graphs used. In the third category we propose a new attribute called *Product of node degree entropy (PNE)* which is based on *degree entropy*, a multiplex property proposed by F. Battistion et al. (Battiston et al., 2013). If degree of node u is $k(u)$, the degree entropy is calculated as

$$E(u) = -\sum_{\alpha=1}^{m} \frac{k(u)^{[\alpha]}}{k_{total}} \log(\frac{k(u)^{[\alpha]}}{k_{total}})$$

where $k_{total} = \sum_{\alpha=1}^{m} k(u)^{[\alpha]}$ and $PNE(u, v) = E(u) * E(v)$ We also extend the same concept to define entropy of a simple topological attribute, say X_{ent}

$$X_{ent}(u, v) = -\sum_{\alpha=1}^{m} \frac{X(u, v)^{[\alpha]}}{X_{total}} \log(\frac{X(u, v)^{[\alpha]}}{X_{total}})$$

where $X_{total} = \sum_{\alpha=1}^{m} X(u, v)^{[\alpha]}$. The entropy based attributes are more suitable to capture the distribution of the attribute value over all dimensions. A higher value indicates uniform

distribution attribute value across the multiplex layers. We denote average and entropy based attributes as *multiplex attributes*.

3 Experiments

We evaluated our approach using data obtained from DBLP [1] databases of which we created three datasets, each corresponding to a different period of time. Table.1 summarizes the information about the graphs of each dataset. Each graph has four years for learning or training and next two years are used to label the examples generated from the learning graphs. Examples are unconnected node pairs and they are labelled as *positive* or *negative* based on whether they are connected during the labelling period or not. Table.2 shows the number of examples obtained for each dataset.

Years	Properties	Co-Author	Co-Venue	Co-Citation
1970-1973	*Nodes*	91	91	91
	Edges	116	1256	171
1972-1975	*Nodes*	221	221	221
	Edges	319	5098	706
1974-1977	*Nodes*	323	323	323
	Edges	451	9831	993

TAB. 1 – *Graphs*

Years		# Positive	# Negatives
Train/Test	Labeling		
1970-1973	1974-1975	16	1810
1972-1975	1976-1977	49	12141
1974-1977	1978-1979	93	26223

TAB. 2 – *Examples from co-authorship graph*

We selected the following topological attributes : Number of common neighbors (CN), Jaccard coefficient (JC), Preferential attachment (PA) (Huang et al., 2005), Adamic Adar coefficient (AA)(Adamic et Adar, 2003), Resource allocation (RA) (Zhou et al., 2009) and Shortest path length (SPL). We applied decision tree algorithm on one dataset to generate a model and then tested it on another dataset. We are using data mining tool Orange [2] for that. We use four types of combinations of the attributes creating five different sets namely : Set_{direct}(attributes computed only in the co-authorship graph) ; $Set_{direct+indirect}$(attributes computed in co-authorship, co-venue and co-citation graphs) ; $Set_{direct+multiplex}$(attributes computed from co-authorship graph with average attributes obtained from three dimension graphs, and also entropy based attributes) ; Set_{all}(attributes computed in co-authorship, co-venue and co-citation graphs, with average of the attributes, and also entropy based attributes)

1. http ://www.dblp.org
2. http ://orange.biolab.si

and $Set_{multiplex}$(average attributes and entropy based attributes). Table.3 shows the result obtained in terms of F1-measure and area under the ROC curve (AUC). We can see that there is improvement in the F1-measure when we use multiplex attributes. AUC is better for all the sets that include multiplex and indirect attributes for both datasets.

Attributes	**Learning :1970-1973 Test :1972-1975**		**Learning :1972-1975 Test :1974-1977**	
	F-measure	**AUC**	**F-measure**	**AUC**
Set_{direct}	0.0357	0.5263	0.0168	0.4955
$Set_{direct+indirect}$	0.0256	0.5372	0.0150	0.5132
$Set_{direct+multiplex}$	0.0592	0.5374	0.0122	0.5108
Set_{all}	0.0153	0.5361	0.0171	0.5555
$Set_{multiplex}$	0.0374	0.5181	0.0185	0.5485

TAB. 3 – *Results of decision tree algorithm*

4 Conclusion

This paper presents our new approach of link prediction in multiplex networks. We propose some new and extended topological features that can be used for characterizing the unlinked node pairs for link prediction task, including also multiplex relation information. They can be applied to predict links in any of the layers of the network. We tested our method for prediction of co-authorship links on datasets obtained from DBLP databases. The preliminary results show that addition of multiplex information indeed improve the prediction performance and thereby motivates us to continue further our research to confirm this concept further.

Références

Adamic, L. et E. Adar (2003). Friends and neighbors on the Web. *Social Networks 25*(3), 211–230.

Battiston, F., V. Nicosia, et V. Latora (2013). Metrics for the analysis of multiplex networks.

Benchettara, N., R. Kanawati, et C. Rouveirol (2010). A supervised machine learning link prediction approach for academic collaboration recommendation. In *Proceedings of the fourth ACM conference on Recommender systems - RecSys '10*, New York, New York, USA, pp. 253. ACM Press.

Berlingerio, M., M. Coscia, F. Giannotti, A. Monreale, et D. Pedreschi (2011). Foundations of Multidimensional Network Analysis. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pp. 485–489. IEEE.

Hasan, M. A., V. Chaoji, S. Salem, et M. Zaki (2006). Link Prediction using Supervised Learning. In *SIAM Workshop on Link Analysis, Counterterrorism and Security with SIAM Data Mining Conference*, Bethesda, MD.

Huang, Z., X. Li, et H. Chen (2005). Link prediction approach to collaborative filtering. In M. Marlino, T. Sumner, et F. M. S. III (Eds.), *JCDL*, pp. 141–142. ACM.

Liben-Nowell, D. et J. M. Kleinberg (2007). The link-prediction problem for social networks. *JASIST 58*(7), 1019–1031.

Yizhou Sun, Rick Barber, Manish Gupta, Charu C. Aggarwa, et Jiawei Han (2011). Co-Author Relationship Prediction in Heterogeneous Bibliographic Networks. In *Advances on social network Analysis and mining (ASONAM)*, Kaohsiung, Taiwan.

Zhou, T., L. Lü, et Y.-C. Zhang (2009). Predicting missing links via local information. *Eur. Phys. J. B 71*, 623.

Summary

In this work we present a new approach for co-authorship link prediction based on leveraging information contained in general bibliographical multiplex networks. A multiplex network, also called multi-slice or multi-relational network, that we consider here is a graph defined over a set of nodes linked by different types of relations. For instance, the multiplex network we are studying here is defined as follows : nodes represent authors and links can be one of the following types : co-authorship links, co-venue attending links and co-citing links. Other types of links can also be considered involving bibliographical coupling, research theme sharing and so on. We show here a new approach for exploring the multiplex relations to predict future collaboration (co-authorship links) among authors. The applied approach is a supervised-machine learning approach where we attempt to learn a model for link formation based on a set of topological attributes describing both positive and negative examples. While such an approach has been successfully applied in the context on simple networks, different options can be applied to extend it to the multiplex network context. One option is to compute topological attributes in each layer of the multiplex. Another one is to compute directly new multiplex-based attributes quantifying the multiplex nature of dyads (potential links). We show our first results on experiments conducted on real datasets extracted from the famous bibliographical database DBLP that has been enriched with paper citation information.

Index

EGC 2014

Organisateurs

Sponsors