Overlapping Community Extraction: A Link Hypergraph Partitioning based Method

Haicheng Tao[†], Zhiang Wu[*‡], Jin Shi[§], Jie Cao[‡] and Xiaofeng Yu[¶]
[†]*College of Computer Sci. and Eng., Nanjing University of Science and Technology, China*
[‡]*Jiangsu Provincial Key Lab. of E-Business, Nanjing University of Finance and Economics, China*
[*]*Corresponding author: zawuster@gmail.com*
[§]*School of Information Management, Nanjing University, China*
[¶]*School of Business, Nanjing University, China*

***Abstract*—Real-world networks often contain communities with pervasive overlaps such that nodes simultaneously belong to several groups. Community extraction, emerging in recent years, is considered to be a promising solution for finding meaningful communities from social networks. In this paper, we explore overlapping community extraction from a link partitioning perspective. First, we define the local link structure composed of a set of closely interrelated links, by extending the similarity of link-pairs to that of a group of links. Second, based upon our prior work, we transform the problem of mining local link structures into a pattern mining problem, and thus present an efficient mining algorithm. Third, we propose to use the hypergraph to assemble all local link structures, and employ hMETIS for hypergraph partitioning. Finally, based on extracted link communities, we restore the membership of nodes in the original graph owing to its links. Experimental results on various real-life social networks validate the effectiveness of the proposed method.**

***Keywords*-Social Network; Overlapping Community Extraction; Link Partitioning; Local Structure; Hypergraph**

I. Introduction

The last decade has witnessed a great deal of Web applications exhibiting social elements reported in people's daily life. Social computing introduces a novel dimension to the Web that goes beyond connecting HTML pages, resources, services, etc, but emphasizes on connecting users. Such major shift in developing Web applications permits to accommodate users' needs, requirements, and relationships better. Despite the big challenge to integrate social computing with services computing, some research have been undertaken towards this direction [1], [2].

A network community typically refers to a tightly-knit group of actors with many connections between the group members and relatively few connections between groups. Particularly, network communities exhibit specific semantics in different contexts such as people share common interests and keep more contacts, or clusters of HTML pages/Web Services related to common topics/functions. There have been a great deal of efforts devoted to community detection, including both disjoint and overlapping communities [3], [4], [5]. In real life, a person commonly has connections to multiple social groups such as scientific activities, family, friends, and hobbies. Driven by this, overlapping community detection has gained increasing interests, and it aims to discover communities that are not necessarily disjoint [6], [7], [8].

Most of the community detection methods attempt to partition the entire network into a certain number of crisp or overlapping communities. This "partitioning" view of community detection is often impropriate, since a real-life network might probably contain lots of nodes that have weak connections to the communities. In such cases, these partitioning models typically split up weakly connected nodes and group them together with tight communities, which finally impedes us from finding the genuine communities. Recently, the concept of community extraction [9], [10] is proposed to deal with the dilemma of the partitioning view. It aims to extract genuine communities by dropping the weakly connected and unimportant nodes. However, community extraction has been implicitly adopted by many overlapping community detection methods, such as CPM [6], Link [7], GCE [11], MOSES [12], etc. They often discover communities by expanding/merging seed communities (e.g., clique). So, not all nodes are included into cliques, which leads to the fact that the resulting communities cannot cover all of nodes in the network.

This paper explores the overlapping community extraction from a link partitioning perspective. The community of links is preferable to that of nodes since the link is likely to have a unique position whereas the node tends to have multiple positions [7]. Moreover, with disjoint link communities, it is easier to restore the membership of every node owing to its links. In this paper, a bottom-up strategy is proposed for extracting link communities. That is, we start by defining the local link structure composed of a set of closely interrelated links, by extending the similarity of link-pairs to that of a group of links. Based upon our prior work [10], [13], we transform the problem of mining local link structures into a pattern mining problem, and thus present an efficient mining algorithm. To discover disjoint link communities, we propose to use the hypergraph to assemble all local

978-1-4799-5066-9/14 $31.00 © 2014 IEEE
DOI 10.1109/SCC.2014.25

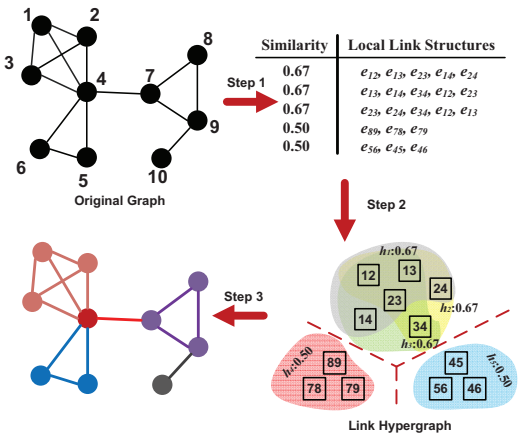


Figure 1: The procedure overview.

link structures, and employ `hMETIS` [14] for hypergraph partitioning.

The remainder of this paper is organized as follows. In Section II, we introduce some preliminaries on overlapping community extraction and an overview of our method. In Section III, we define the local link structure and propose the mining algorithm. In Section IV, we present the definition and partitioning method of link hypergraph, and membership translation based on link communities. Experimental results will be given in Section V. We summarize the related work in Section VI, and finally conclude this paper in Section VII.

II. Preliminaries and Overview

Given a network or graph $G = \{V, E\}$, V is a set of n nodes and E is a set of m edges. In this paper, we limited our scope to the unweighted network which is often determined by a $n \times n$ adjacency matrix $\mathbf{A} = [A_{pq}]$, where $A_{pq} = 1$ $(p \neq q)$ if there is an edge between node i_p and i_q and 0 otherwise. The goal of overlapping community extraction is to seek a good K-way fuzzy partition $\pi = \{C_1, \cdots, C_K\}$, where $C_{k'}$ is the k'th community, and $C_1 \bigcup \cdots \bigcup C_K \subseteq V$. A $n \times K$ membership matrix $\mathbf{U} = [u_{pk'}]$ is used to represent multiple positions of every node. More formally,

$$0 \leq u_{pk'} \leq 1, \ \forall C_{k'} \in \pi, \ \sum_{k'=1}^{K} u_{pk'} = 1 \ \forall i_p \in V. \quad (1)$$

Recently, some attentions have been paid to overlapping community detection based on link clustering [7], [15], [16], of which the underlying assumption is the link has a unique position whereas the node naturally occupies multiple positions owing to its links. Along this line, the proposed method consists of three main steps.

- Step 1: mining local structures each of which is composed of a set of closely interrelated links.
- Step 2: assembling every local structures to form a link hypergraph and thus partitioning the link hypergraph to identify the unique position of every link.
- Step 3: restoring fuzzy membership of nodes according to their links.

Example. Fig. 1 shows the three steps of our procedure in a simple graph. Step 1 mined five local link structures, as shown in upper-right table of Fig. 1. A link hypergraph and its partitions are then generated in Step 2. In Step 2, by restoring membership back to the original graph, three communities are identified with overlap at node #4. Note that link $e_{9,10}$ has not been included by any local link structure, which leads to removal of the weak-tie node #10 from all communities.

III. Local Link Structures Mining

This section will focus on the Step 1 of our method: local link structures mining. We firstly present the definition and its implications of local link structures, and then briefly introduce algorithmic details of CoPaMï, a novel FP-growth-like [17] COsine PAttern MIning algorithm [10], [18].

A. Definition and the Beyond

In the literature [7], a hierarchical clustering with a similarity between links was used to build a dendrogram, and thus link communities were extracted by cutting this dendrogram based upon the heuristic metric named *partition density*. Each node inherits all memberships of its links and can thus belong to multiple communities. So, to define the similarity of a link pair becomes the fundamental issue. Let e_{pr} denote a link connecting nodes i_p and i_r (i.e., *impost nodes*). Given links e_{pr} and e_{qr} sharing a node i_r (i.e., *keystone*), the the Jaccard index can be defined as [7]

$$Jaccard(e_{pr}, e_{qr}) = \frac{|N_p \bigcap N_q|}{|N_p \bigcup N_q|}, \quad (2)$$

where N_p is the set of neighbors (i.e., friends) of i_p. This definition indicates that the similarity of link pairs is *irrelevant to* the keystone but only depends on the set of impost nodes.

In this paper, we use the *cosine* as similarity measure instead of the Jaccard index. The reason lies in that cosine holds the so-called Ordered Anti-Monotone Property (OAMP) [10], [18] which gives birth to a high-efficient algorithm for mining local link structures. We define the cosine similarity between e_{pr} and e_{qr} as

$$\cos(e_{pr}, e_{qr}) = \frac{|N_p \bigcap N_q|}{\sqrt{|N_p||N_q|}}. \quad (3)$$

Furthermore, Eq. (3) can be naturally extended to multiple links case. Let $h = \{e_{1r}, \cdots, e_{|h|r}\}$ be a link set of which the keystone is i_r. The cosine of h is defined by

$$\cos(h) = \frac{|N_1 \bigcap \cdots \bigcap N_{|h|}|}{\sqrt[|h|]{\prod_{p=1}^{|h|} |N_p|}}. \quad (4)$$

If $\cos(h) \geq t_c^*$ where $0 \leq t_c^* \leq 1$ is a predefined threshold, all links in h are closely interrelated. We further note that the link similarity is actually determined by the set of impost nodes. For example, h can be constructed by the set of impost nodes $S = \{i_1, \cdots, i_{|h|}\}$ which is a close-knit group of nodes in essence. To understand this, let $t_c^* = 1$ which means $N_1 = \cdots = N_{|h|}$, and thus S are *equivalent structures* [19]. As t_c^* increases, the impost node set S becomes looser than the equivalent structure, but S is still a close-knit structure w.r.t. t_c^*.

Since a local link structure h is essentially determined by the set of impost nodes S, if all of such sets $\mathcal{S} = \{S | \cos(S) = \cos(h) \geq t_c^*\}$ in graph G are mined, the set of local link structures $\mathcal{H} = \{h | \cos(h) \geq t_c^*\}$ can be derived by expanding links between impost nodes and keystones.

B. Algorithm Design

Consider the mining problem of the impost node sets $\mathcal{S}$. Somewhat to one's surprise, this mining task can be proven to be equivalent to *cosine interesting pattern mining* in the realm of traditional association rule analysis. To understand this, we first represent the network G using the *transaction model* as $\mathcal{D}$, where each line corresponds to a node and items in this line are its neighbors, i.e., $\forall T_p \in \mathcal{D}, T_p = N_p$. A formal definition for the cosine interesting patterns (itemsets) can be given as follows [18]:

Definition 1 (Cosine Interesting Patterns)**.** *Let $\mathcal{D}$ be a transaction database over a universal itemset $\mathcal{I}$, min_supp be the minimum support threshold, and min_cos be the minimum cosine threshold. The collection of the cosine interesting patterns in $\mathcal{D}$ w.r.t. min_supp and min_cos are defined by $\mathcal{F}(\mathcal{D}, min\_supp, min\_cos) = \{X \subseteq \mathcal{I} | supp(X) \geq min\_supp, \cos(X) \geq min\_cos\}$.*

In the realm of social networks, $supp(S) = \frac{|N_1 \bigcap \cdots \bigcap N_{|h|}|}{n}$ indicates the proportion of common friends (i.e., neighbors) in S. Moreover, if we set $min\_supp = t_s^* = 0$ and $min\_cos = t_c^*$, mining $\mathcal{S}$ is equivalent to mining cosine interesting patterns in Definition 1. Therefore, each impost node set $S \in \mathcal{S}$ is a cosine pattern in essence.

An efficient algorithm named CoPaMï for mining cosine patterns was presented in [10], [18]. One of the distinguished features of CoPaMï can employ the cosine measure work as support to prune uninteresting itemsets in advance. As is known, the anti-monotone property (AMP) held by the support measure determines it can be leveraged to reduce the search space of frequent patterns. However, the cosine measure does not hold the AMP. Fortunately, we found an alternative to AMP for employing cosine as a pruning measure. This alternative property is named *Ordered Anti-Monotone Property* (OAMP) which is defined as follows:

Algorithm 1 CoPaMï

```
1: Create the FP-tree Tree for transaction data D;
2: Let S store the current suffix, S ← ∅;
3: Let S store the cosine patterns (i.e., core close-knit structures), S ← ∅;
4: procedure CP-GROWTH(Tree, X, t_s^*, t_c^*, S)
5:     for each item i_k from bottom to top in Tree's head table do
6:         generate candidate pattern S' ← {i_k} ∪ S;
7:         if cos(S') ≥ t_c^* then                    ▷ cos(S') = 1 if |S'| = 1
8:             S ← S_c ∪ {S'};
9:             create the conditional FP-tree Tree_{S'} for S';
10:            CP-GROWTH(Tree_{S'}, S', t_s^*, t_c^*, S);
11:        end if
12:    end for
13:    return S;
14: end procedure
```

Definition 2 (OAMP)**.** *Let I be a universal itemset. A measure M holds the Ordered Anti-Monotone, if $\forall\ S, S' \subseteq I$, given that (1) $S \subseteq S'$, and (2) if $S' \setminus S \neq \emptyset$, $\forall\ i_p \in S$ and $i_{p'} \in S' \setminus S$, $s(\{i_p\}) \leq s(\{i_{p'}\})$, we have $M(S) \geq M(S')$.*

Definition 2 implies that OAMP can be regarded as a special case of AMP. That is, a measure possessing AM certainly holds OAMP, such as the support measure; but the reverse is not true — cosine similarity is just an example. Compared with the well-known AMP, OAMP demands an extra condition that all the items in the difference set ($S' \setminus S$) must have higher supports than the items in the subset (S). We then have the following theorem:

Theorem 1. *Cosine similarity holds the ordered anti-monotone.*

Due to the limited space, we remove the proof and refer the readers for details in [10]. We here present the pseudocodes of CoPaMï in Algorithm 1, and provide some briefly explanations. Note first that to utilize the OAMP of the cosine measure, CoPaMï demands the FP-tree to be built on transactions with items sorted in a *support-descending* order. It is also noteworthy that CP-growth no longer distinguishes between single-path trees and multi-path trees, which were however treated differently in the classic FP-growth procedure. In FP-growth, when a conditional FP-tree is a single-path tree, FP-growth can simply enumerate all the node combinations as frequent itemsets without further sub-tree projections. This should be attributed to the powerful AMP. However, since the cosine measure only possesses OAMP, CP-growth has to continue the sub-tree projection as for the multi-path trees, although in a much simpler manner — a single-path tree will be always projected to a single-path sub-tree. The pseudocodes in Line 10 of Algorithm 1 are for the sub-tree projections, for both single-path and multi-path trees.

Each cosine pattern forms the backbone of a local link structure. The Algorithm 2 is then invoked for link expansion. Generally, two kinds of links associated with a

Algorithm 2 Link Expansion

Let $\mathcal{H}$ store local link structures derived from $\mathcal{S}$;
procedure LINK($\mathbf{A}$, $\mathcal{S}$)
  for each element S_j in $\mathcal{S}$ **do**
    Find all keystones of S_j, denoted as R_j;
    $\forall A_{pq} = 1, i_p \in S_j, i_q \in S_j \bigcup R_j$, $h_j = h_j \bigcup e_{pq}$;
  end for
  return $\mathcal{H} = \bigcup_j h_j$;
end procedure

cosine pattern should be expanded. That is, links within the pattern, and links between keystones and every node. Given a cosine pattern S, the number of keystones is $\sigma(S) = |N_1 \bigcap \cdots \bigcap N_{|h|}|$, i.e., S's support count. Therefore, besides the inner edges in S, there are $\sigma(S) \cdot |h|$ links between keystones and every node.

IV. Link Hypergraph Partitioning

Given a large number of resulting local link structures, one problem is arising. That is, how can we easily merge adjacent/relevant link structures to obtain a set of crisp link communities? The link graph [15], [16] is a potential model for helping to identify link communities. Each node of a link graph is an edge of the original graph, and nodes are connected if the corresponding links in the original graph are connected through some node. Obviously, the link graph can only capture interrelated *link-pairs*. However, the local link structure in this paper contains a set of interrelated links. So, it is naturally to extend common graph to *hypergraph* for assembling local link structures.

In mathematics, a hypergraph is a generalization of a graph in which an edge can connect any number of vertices. So the hypergraph is a natural choice for assembling local link structures to form a generalized link graph. More precisely, we define link hypergraph as follows.

Definition 3 (Link Hypergraph)**.** *A link hypergraph $HG = \{E', \mathcal{H}, \omega\}$. E' is the set of vertices, and it is a subset of edge set in the original graph $G = \{V, E\}$, i.e., $E' \subseteq E$. $\mathcal{H}$ is the set of hyperedges and ω is the weight of every hyperedge. Each hyperedge $h_j \in \mathcal{H}$ corresponds to a local link structure, and all of edges of G in h_j are regarded as vertices in HG and they are connected by h_j. The weight of the hyperedge h_j is the cosine similarity of this local link structure, i.e., $\omega_j = \cos(h_j)$.*

Example. We can also use the toy example in Fig. 1 to illustrate the link hypergraph. Five local link structures are mined from the original graph. A link hypergraph is then constructed as shown in the bottom-right of Fig. 1, and it contains five hyperedges h_1 to h_5. Specifically, h_1, h_2 and h_3 were coupled together and totally connected six vertices. h_4 and h_5 formed two separate parts, and each of them connected three vertices.

Hypergraph Partitioning. Then, the problem is how to partition the link hypergraph into K disjoint parts, and the unique position of every node in hypergraph, i.e., link in the original graph, can thus be identified. Hypergraph partitioning is an important problem attracting a great many attentions from numerous areas, including VLSI design, transportation management, and data mining. In this paper, we employ the software package `hMETIS` [14] for partitioning the link hypergraphs.

Membership Translation. Most fuzzy community methods require membership weights quantifying how strongly a node belongs to a particular community. This final step restores the membership of a node from the result of link partitioning. Simply, the membership of a link is mapped to that of its endpoints. A fuzzy community membership of a node can be computed by counting the number of link membership a node has.

Table I: Real-world networks for experiments.

Name	$\|V\|$	$\|E\|$	$< k >$	C
`LesMis`	77	254	6.60	0.736
`Google`	944	1611	3.24	0.570
`Protein`	2614	6379	4.88	0.299
`Words`	7194	31771	8.83	0.206

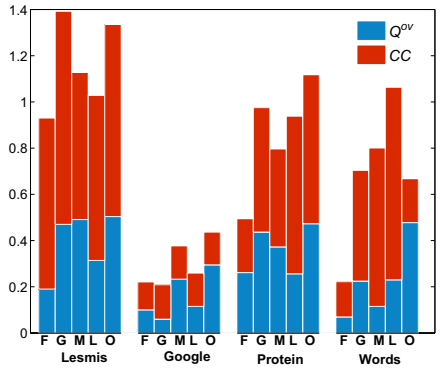
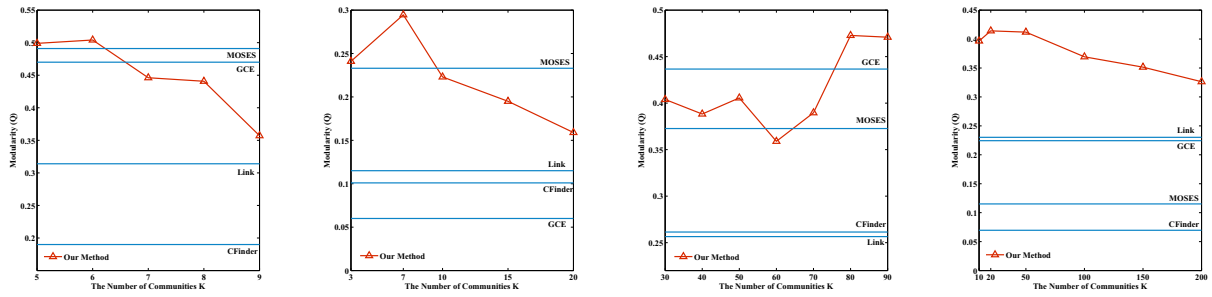


Figure 2: Overall performance comparison.

V. Experimental Validation

In this section, we present experimental results on four real-world networks including `LesMis`, `Google`, `Protein` and `Words`. Some characteristics of these data sets are shown in Table I, where $|V|$ and $|E|$ indicate the numbers of nodes and edges respectively in the network, $< k >= 2|E|/|V|$ indicates the average degree, and C indicates the average clustering coefficient. `LesMis` is the network of coappearances of characters in Victor Hugo's novel, where nodes represent characters and edges connect any pair of characters that appear in the same chapter of the book. `Google` is a directed graph where each node represents Google web pages. `Protein` describes the protein-protein interactions. `Words` is a network collected by participants who wrote the first word that came to mind that was associated to the given word.



(a) LesMis (b) Google (c) Protein (d) Words

Figure 3: The impact of K.

A. Comparison Results

Other Tools. Four overlapping community detection tools, i.e., CFinder (F) [6], Link (L) [7], GCE (G) [11], and MOSES (M) [12], were used in the experiments for the purpose of comparison. Note that the existing overlapping community detection methods were roughly categorized into four classes [4] (e.g., clique percolation, link partitioning, local expansion and optimization, and fuzzy detection). From this, we carefully selected these four tools, as is, one tool from one category. In particular, CFinder (http://www.cfinder.org/) is the implementation of the clique percolation method (CPM) [6], of which the unique parameter k indicates the size of seed k-cliques. We set k=4 in the experiment. Link method (https://github.com/bagrow/linkcomm) employs hierarchical clustering to build a link dendrogram and then cut this dendrogram according to the partition density D [7]. We report the result of link method with the maximum D for each network. Similar to CFinder, GCE (https://sites.google.com/site/greedycliqueexpansion) also mines k-cliques as seeds and expands each seed to a community until the predefined fitness function is locally maximized [11]. So, k=4 is also the default setting. MOSES (https://sites.google.com/site/aaronmcdaid/moses) combines the stochastic block model and the local optimization scheme for fuzzy detection [12], and one advantage of MOSES is no parameters are required as input. For our proposed method (O), we set $t_s^* = 2$ and $t_c^* = 0.5$, and K varies with different networks.

Evaluation Metrics. When the ground-truth community is unknown, we utilize two factors, i.e., quality and coverage, together to evaluate the performance of identified communities. For the quality factor, the *modularity* is one of the most widely-used measures [4], [3]. Thus, we adopt *overlapping modularity* Q^{ov} as the validation measure. It is computed as follows:

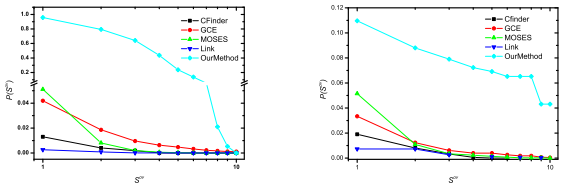
$$Q^{ov} = \frac{1}{2|E|} \sum_{k'=1}^{K} \sum_{p,q \in C_{k'}} [A_{pq} - \frac{|N_p||N_q|}{2|E|}] u_{pk'} u_{qk'}, \quad (5)$$

where $|N_p|$ is the degree of node i_p, and $u_{pk'}$ is the membership of node i_p in the community $C_{k'}$. The value of Q^{ov} is in the interval: (-1,1), and a larger value indicates a better quality. For the coverage factor, we employ community coverage (CC in short) presented in [7]. To compute CC, we simply count the fraction of nodes that belong to at least one community. Thus, a larger CC implies more nodes are extracted as community members.

Results. Fig. 2 shows $Q^{ov} + CC$ values of five methods on four real-world networks. For our method, the best results were reserved after trying different K values . As can be seen, our method shows the best Q^{ov} in every network, which implies the quality of overlapping communities discovered by our method is higher than other tools. Meanwhile, indicated by CC, our method extracts much more nodes in most networks, except on Words. Overall, this result demonstrates that our method can extract high-quality communities for a large fraction of real-world networks.

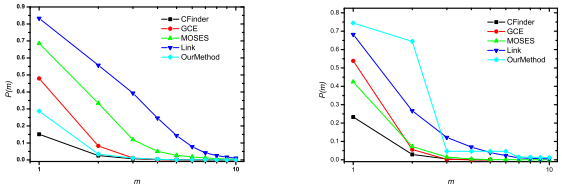
An important parameter, the number of communities K, is fixed by other four tools, since K is commonly determined by the pre-defined seed structures and the merging rule. For instance, CFinder mined a certain number of k-cliques as seeds, and then merged adjacent k-cliques sharing $k-1$ nodes, which causes K is out of control. Similar cases happened in Link, GCE, and MOSES. We therefore investigate the impact of the parameter K in our method. As can be seen from Fig. 3, our method is more stable than other tools such that at least two points are higher than the best one among four tools in every network. Meanwhile, the gap narrows in the small network LesMis but becomes wider in the other three large networks. Interestingly, we observed that the ranking of selected four tools varies largely as different networks, which implied they are more network-dependent.

What about the overlapping degree of identified communities? To answer this, we compare overlapping communities discovered by five methods on two cumulative distributions. Let S^{ov} denote the overlap size between any two communities, and m be the number of communities that



(a) Words (b) Protein

Figure 4: The cumulative distribution of S^{ov}.



(a) Words (b) Protein

Figure 5: The cumulative distribution of m.

a node belongs to. We construct two cumulative distribution functions $P(S^{ov})$ and $P(m)$ indicating the proportion of variables that are larger than S^{ov} and m, respectively. Note that these two functions were also used in [6], [7] to examine the overlapping degree of communities.

Fig. 4 and Fig. 5 depict $P(S^{ov})$ and $P(m)$ of five tools on two networks Words and Protein. Overall, our method can successfully capture multiple relationships and a great deal of overlap. One exception occurred in Fig. 5(a), that is, $P(m)$ of our method on Words is obviously low. This is because our method only extracted a small portion of core nodes, i.e., a lower CC value. However, $P(S^{ov})$ of our method on Words is much higher, which implies overlapping nodes are concentrated in some few communities.

Finally, we investigate the efficiency of our method which mainly consists of two phases: local link structures mining and hypergraph partitioning by using hMETIS. Since the multilevel hypergraph partitioning algorithm used in hMETIS is quite efficient, we here show the efficiency for mining local link structures. Fig. 6 depicts the results on four networks by setting t_c^* =0.5, 0.55, and 0.6 respectively. In general, our method is rather high-efficient (e.g., less than 7s for all networks), which reflects the powerful pruning effect of cosine similarity in CoPaMï. Obviously, the higher t_c^* is, the faster the execution time is. We leave for further research on applying our method towards super-large scale networks. Though the scale of Google is moderate, it consumed the most execution time among four networks. This is due in large part to the sparsity of Google network.

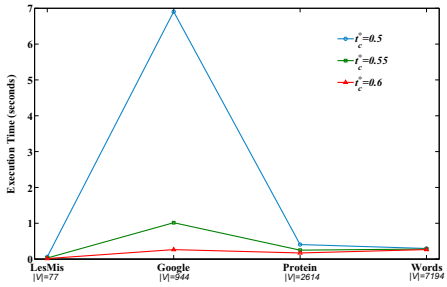


Figure 6: The efficiency for mining local link structures.

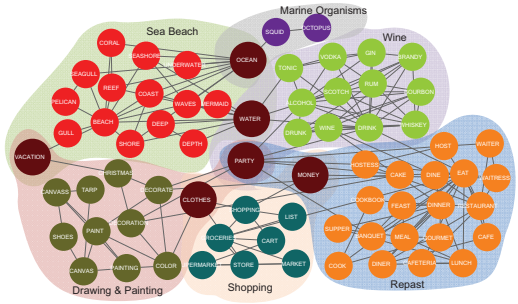


Figure 7: Sampling communities from the Word network.

B. Qualitative Evaluation

The Q^{ov} and CC only give us a general picture about the performance of community detection. For some complex networks, such as Words network, a high Q^{ov} but low CC is not enough to show the identified communities are truly meaningful. For that, we here sample several communities from Words to observe their semantics. We set $K = 60$ to obtain relatively smaller communities, and sampled six communities as shown in Fig. 7.

We can find from Fig. 7 that words in the same community are closely-related with a certain topic. According to that, we named six communities as “Sea Beach”, “Marine Organisms”, “Wine”, “Repast”, “Shopping” and “Drawing & Painting”. Especially, the overlapping nodes are marked as dark brown larger circles. The semantic meanings of these overlapping nodes are apparently explainable. For example, “water” belongs to “Sea Beach” and “Wine” communities, and “money” belongs to three communities including “Repast”, “Shopping” and “Drawing & Painting”. More interestingly, if we were asked to recommend something related to “money”, things in “Repast” community are more suitable according to the size of communities, since food is always a major concern of anyone.

VI. Related Work

Community detection is a ceaselessly fundamental problem in network science. There have been a large body of existing work dealing with disjoint community detection.

Most of these studies can be classified into two main categories, in terms of whether or not explicit optimization objectives are being used. The methods with global models typically consider the global topology of a network, and aim to optimize a criterion defined over a network partition. Some methods along this line include the Kernighan-Lin algorithm [20], latent space models, stochastic block models, modularity optimization [21], and traditional clustering techniques such as K-means, multi-dimensional scaling, and spectral clustering. The differences between these methods ultimately come down to the precise definition of a "denser" community, i.e., the global criterion and the algorithmic heuristic followed to identify such sets.

The methods without global models typically employ a bottom-up strategy to find communities. They often start by defining the properties of a node, a pair of nodes, or a group of nodes in a same community, and then search within a whole network for the communities that hold the proposed properties [22]. A network's global community structure is detected by considering the ensemble of communities obtained by looping over all of these local structures. A community could be regarded as a clique, a k-club, a quasi-clique, or the combination of node pairs that have nodes similar to each other [19]. Our method falls into this category, i.e., a bottom-up strategy without global models, but it aims to discover *link communities* instead.

Due to the pervasive overlaps among communities in real-life network, there is therefore a clear need to develop overlapping community detection/extraction methods. According to the latest review [4], methods for overlapping community detection roughly fall into four categories, i.e., clique percolation, link partitioning, local expansion and optimization, and fuzzy detection. Clique percolation originated from CPM (e.g., CFinder) [6], though several extension on CPM have appeared including CPMw [23] and SCP [24]. Ahn et al. first proposed the idea of link partitioning [7], after which many methods were developed along this line [16], [25]. Local expansion and optimization algorithms usually used close-knit structures as seeds, and then expanded them based on a local benefit function. There are many methods of this kind including GCE [11], LFM [8], EAGLE [26], OSLOM [27], etc, among which GCE is the most famous one. Fuzzy detection is similar to fuzzy c-means (FCM), which modeled the problem as an optimization problem and solved it to obtain a soft membership vector for every node. Lots of matured models were used in fuzzy detection. For instance, FCM was used in [28], mixture models were used in [29], non-negative matrix factorization was used in [30], [31] and the stochastic block model (SBM) was used in [32]. MOSES is a tool based on SBM.

VII. Conclusion

This paper presented a novel method for overlapping communities extraction, which enables us to extract link communities by using a bottom-up strategy and hypergraph partitioning method. We start by extending the similarity of link-pairs to that of a group of links in order to define the local link structure. Thus, mining local link structures were transformed into a pattern mining problem, and an FP-growth-like mining algorithm was presented. Given a large number of local link structures, we propose to use the hypergraph to assemble all local link structures, and employ `hMETIS` for hypergraph partitioning to discover disjoint link communities. Experimental results on four real-life networks demonstrated that our method outperformed the state-of-art methods on the quality of identified communities while reserving comparable community coverage. In particular, the semantics of extracted communities shows our method can extract meaningful communities with rational overlapping nodes.

Acknowledgment

This research was partially supported by the National Natural Science Foundation of China (NSFC) under Grants 61103229, 71372188 and 61100197, National Center for International Joint Research on E-Business Information Processing under Grant 2013B01035, National Key Technologies R&D Program of China under Grant 2013BAH16F03, International S&T Cooperation Program of China under Grant 2011DFA12910, National Soft Science Research Program under Grant 2013GXS4B081, Industry Projects in Jiangsu S&T Pillar Program under Grant BE2012185, and Key Project of Natural Science Research in Jiangsu Provincial Colleges and Universities under Grant 12KJA520001.

References

[1] K. Chard, K. Bubendorfer, S. Caton, and O. F. Rana, "Social cloud computing: A vision for socially motivated resource sharing," *IEEE Transactions on Services Computing*, vol. 5, no. 4, pp. 551–563, 2012.

[2] A. Mohaisen, "Socialcloud: Using social networks to build distributed computing services," University of Minnesota, Tech. Rep., 2011.

[3] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, pp. 75–174, 2010.

[4] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: the state of the art and comparative study," *ACM Computing Surveys*, vol. 45, no. 4, 2013.

[5] T. Qian, Q. Li, J. Srivastava, Z. Peng, Y. Yang, and S. Wang, "Exploiting small world property for network clustering," *World Wide Web*, vol. 17, no. 3, pp. 405–425, 2014.

[6] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, pp. 814–818, 2005.

[7] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[8] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, no. 3, p. 033015, 2009.

[9] Y. Zhao, E. Levina, and J. Zhu, "Community extraction for social networks," *Proceedings of the National Academy of Sciences of the USA (PNAS)*, vol. 108, no. 18, pp. 7371–7326, 2011.

[10] Z. Wu, J. Cao, J. Wu, Y. Wang, and C. Liu, "Detecting genuine communities from large-scale social networks: a pattern-based method," *The Computer Journal*, p. bxt113, 2013.

[11] C. Lee, F. Reid, A. McDaid, and N. Hurley, "Detecting highly overlapping community structure by greedy clique expansion," *arXiv preprint arXiv:1002.1827*, 2010.

[12] A. McDaid and N. Hurley, "Detecting highly overlapping communities with model-based overlapping seed expansion," in *2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2010, pp. 112–119.

[13] Z. Wu and J. Luo, "The measurement model of grid qos," in *The 10th International Conference on Computer Supported Cooperative Work in Design (CSCWD'06)*. IEEE, 2006, pp. 1–6.

[14] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: applications in vlsi domain," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, no. 1, pp. 69–79, 1999.

[15] T. Evans and R. Lambiotte, "Line graphs, link partitions, and overlapping communities," *Physical Review E*, vol. 80, no. 1, p. 016105, 2009.

[16] T. Evans and R. Lambiotte, "Line graphs of weighted networks for overlapping communities," *The European Physical Journal B*, vol. 77, no. 2, pp. 265–272, 2010.

[17] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. Dallas, Texas, USA: ACM Press, New York, 16–18 May 2000, pp. 1–12.

[18] J. Cao, Z. Wu, and J. Wu, "Scaling up cosine interesting pattern discovery: A depth-first method," *Information Sciences*, vol. 266, no. 0, pp. 31–46, 2014.

[19] L. Tang and H. Liu, "Community detection and mining in social media," *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 1–137, 2010.

[20] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell Systems Technical Journal*, vol. 49, pp. 291–307, 1970.

[21] M. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, p. 066113, 2004.

[22] L. Tang, X. Wang, and H. Liu, "Community detection via heterogeneous interaction analysis," *Data Mining Knowledge Discovery*, vol. 25, pp. 1–33, 2012.

[23] I. Farkas, D. Ábel, G. Palla, and T. Vicsek, "Weighted network modules," *New Journal of Physics*, vol. 9, no. 6, p. 180, 2007.

[24] J. M. Kumpula, M. Kivelä, K. Kaski, and J. Saramäki, "Sequential algorithm for fast clique percolation," *Physical Review E*, vol. 78, no. 2, p. 026109, 2008.

[25] Y. Kim and H. Jeong, "Map equation for link community," *arXiv preprint arXiv:1105.0257*, 2011.

[26] H. Shen, X. Cheng, K. Cai, and M.-B. Hu, "Detect overlapping and hierarchical community structure in networks," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 8, pp. 1706–1712, 2009.

[27] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, "Finding statistically significant communities in networks," *PloS one*, vol. 6, no. 4, p. e18961, 2011.

[28] S. Zhang, R.-S. Wang, and X.-S. Zhang, "Identification of overlapping community structure in complex networks using fuzzy c-means clustering," *Physica A: Statistical Mechanics and its Applications*, vol. 374, no. 1, pp. 483–490, 2007.

[29] M. Magdon-Ismail and J. Purnell, "Fast overlapping clustering of networks using sampled spectral distance embedding and gmms," *Rensselaer Polytechnic Inst., Tech. Rep*, 2011.

[30] M. Zarei, D. Izadi, and K. A. Samani, "Detecting overlapping community structure of networks based on vertex–vertex correlations," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2009, no. 11, p. P11013, 2009.

[31] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, "Overlapping community detection using bayesian non-negative matrix factorization," *Physical Review E*, vol. 83, no. 6, p. 066114, 2011.

[32] K. Nowicki and T. A. B. Snijders, "Estimation and prediction for stochastic blockstructures," *Journal of the American Statistical Association*, vol. 96, no. 455, pp. 1077–1087, 2001.