Chinese Journal of Electronics
Vol.26, No.2, Mar. 2017

Discovering Overlapping Communities by Clustering Local Link Structures*

TAO Haicheng[1], WANG Youquan[1], WU Zhi'ang[2], BU Zhan[2] and CAO Jie[1,2]

(1. *College of Computer Sci. and Eng., Nanjing University of Science and Technology, Nanjing 210094, China*)

(2. *School of Information Engineering, Nanjing University of Finance and Economics, Nanjing 210003, China*)

Abstract — Recent advances point out that the existing community detection methods commonly face two challenges: incorrect base-structures and incorrect membership of weak-ties. To overcome both problems, a Local link structure (LLS) clustering based method for overlapping community detection is proposed. We extend the similarity of a pair of links to a group of links named LLS, and thus transform mining LLSs as a pattern mining problem. We prove that LLS with an appropriate threshold can filter weak-ties in the form of bridge and local bridge with its span being larger than 3. A compositive framework is presented for overlapping community detection based on LLS mining and clustering. Comparative experiments on both synthetical and real-world networks demonstrate that our method has advantage over six existing methods on discovering higher quality communities.

Key words — Complex network, Overlapping community detection, Weak-ties, Hypergraph, Clustering.

I. Introduction

Community structure provides a significant clue to understand structural and functional properties of a network consists in discovering its communities. In real life, a person commonly has connections to multiple social groups such as scientific activities, family, friends, and hobby. Driven by this, Overlapping community detection (OCD) aims to discover communities that are not necessarily disjoint[1−3].

Recently, two kinds of approaches[2−5] attract particular attentions in the realm of OCD. One is the structure-based methods that target at searching within a whole network for the local structures that hold a pre-defined property, *e.g.*, the Clique percolation method (CPM)[2] searches cliques. Another is the link-clustering methods that aim to detect link communities. Thus, with disjoint link communities, nodes naturally occupy multiple positions owing to their links. In this category, the link method based on hierarchical clustering is the famous one[3]. The proposed method could be viewed as a combination of above approaches.

As reported in Ref.[6], two challenges commonly exist in the current methods. On one hand, for the structure-based methods, the pre-defined base-structure may be incorrect or too strict, so that the graph might not include such base-structures. For example, in CPM, the graph might not include many k-cliques especially when k becomes large. On the other hand, the membership of weak-tie cannot be well handled. A weak-tie had better not be included in any community since it does not represent strong relationship between nodes. However, none of special technique has been designed in current OCD methods to exclude weak-ties from communities.

Motivated by this, we present a novel LLS clustering based method for OCD. We extend the similarity of a pair of links to a group of links named LLS, and reason out that a LLS can be constructed by a cosine pattern. This builds up the connection between structure based and link clustering based methods. We further prove that weak-ties can be excluded in any community, by setting an appropriate similarity threshold when mining LLSs. Thus, a compositive OCD framework is presented which consists of LLS mining, LLS clustering and membership translation.

II. Local Link Structure Mining

Given a network $G = \{V, E\}$, V is a set of n nodes and E is a set of m edges. Our goal is to discover the overlapping community structures, that is, to obtain a

*Manuscript Received Dec. 17, 2014; Accepted Sept. 9, 2015. This work is supported by the National Natural Science Foundation of China (No.71571093, No.71372188, No.61502222), National Center for International Joint Research on E-Business Information Processing (No.2013B01035), and National Key Technologies R&D Program of China (No.2013BAH16F03).

 DOI:10.1049/cje.2017.01.017

$n \times K$ membership matrix $\boldsymbol{U} = [u_{pk}]$ for representing the K-way fuzzy partitioning of G. More formally, $0 \leq u_{pk} \leq 1,\ \forall i_p \in V,\ \sum_{k=1}^{K} u_{pk} = 1$. In what follows, we begin by deriving the definition of LLS, and thus mining and clustering LLS.

1. Definition and the beyond

The basic premise of link clustering based methods for OCD is the link has a *unique* position whereas the node naturally occupies multiple positions owing to its links. Most notably, Ahn *et al.*[3] proposed a Jaccard-type similarity score for a pair of links. Given two links e_{pr} and e_{qr} sharing a node i_r (*i.e.*, *keystone*), the Jaccard index is defined as $\frac{|N_p \bigcap N_q|}{|N_p \bigcup N_q|}$, where N_p is the set of neighbors (i.e., friends) of i_p.

Proposition 1 The similarity of link pair is irrelevant to the keystone (*i.e.*, i_r) but only depends on the neighborhood set of impost nodes (*i.e.*, i_p and i_q).

Here, we replace the Jaccard similarity by the *cosine* similarity, *i.e.*, $\cos(e_{pr}, e_{qr}) = \frac{|N_p \bigcap N_q|}{\sqrt{|N_p||N_q|}}$. To compute the similarity of all link pairs suffers from computational inefficiency, especially when the network involves into large scale, *e.g.*, the number of link pairs is n^2. So, our solution is to define the similarity of a group of links (*i.e.*, LLS) and thus to mine LLS with high-similarity directly. More precisely, let $h = \{e_{1r}, \cdots, e_{|h|r}\}$ be a LLS, where i_r is one of its keystones. Then, the cosine similarity of h is

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

If $\cos(h) \geq t_c^*$ where $0 \leq t_c^* \leq 1$ is a predefined threshold, all links in h are closely interrelated. Based on the basic observation, h is determined by the set of impost nodes $S = \{i_1, \cdots, i_{|h|}\}$ which is a close-knit group of nodes in essence. If we set $t_c^* = 1$, S is an equivalent structure[7]. As the decrease of t_c^*, S becomes looser than the equivalent structure, but S is still a close-knit structure w.r.t. t_c^*. We have:

Proposition 2 Let $\mathcal{I}_S$ be the set of keystones of S. A set of local link structures can be derived from $\mathcal{I}_S$ and S, *i.e.*, $H_S = \{h_1, \cdots, h_s\}, \forall i_r \in \mathcal{I}_S, h_r = \{e_{1r}, \cdots, e_{|h|r}\}$. We have $\cos(S) = \cos(h_r), \forall i_r \in \mathcal{I}_S$.

In other words, a close-knit group of nodes S with $\cos(S) \geq t_c^*$ can construct a set of LLSs with the similarity being greater than t_c^*. That is, after searching all keystones of S, a LLS is composed of links between a keystone and impost nodes.

2. Mining LLS through mining cosine patterns

Proposition 2 implies that mining close-knit LLSs can be realized by mining $\mathcal{S}, \forall S \in \mathcal{S}, \cos(S) \geq t_c^*$. Somewhat to one's surprise, this task can be proven to be equivalent to cosine pattern mining in data mining. To illustrate this, we represent the network G as a transaction database $\mathcal{D}$, 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 patterns (itemsets) is given as follows[8]:

Definition 1 Let $\mathcal{I}$ be the universal itemset of $\mathcal{D}$, and *min_supp* and *min_cos* be the minimum support and cosine threshold, respectively. The collection of the cosine patterns in $\mathcal{D}$ is defined by $\mathcal{F}(\mathcal{D}, min\_supp, min\_cos) = \{X \subseteq \mathcal{I} | supp(X) \geq min\_supp, \cos(X) \geq min\_cos\}$.

Going back to the social network, $supp(S) = \frac{|N_1 \bigcap \cdots \bigcap N_{|h|}|}{n}$ 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 the set $\mathcal{F}$ of cosine patterns according to Definition 1. Therefore, each impost node set $S \in \mathcal{S}$ is a cosine pattern in essence. An efficient algorithm named CoPaMi for mining cosine patterns was presented in our previous work[8,9]. One of the distinguished features of CoPaMi can employ the cosine measure working as support to prune uninteresting itemsets in advance.

How to set parameters t_s^* and t_c^* might become the biggest concern when using CoPaMi. Here, we provide some clues for parameter settings. Since the cosine similarity is a relative measure, we commonly set $t_c^* \in [0.5, 0.6]$ to guarantee the cohesiveness of a LLS. Furthermore, we find when $t_s^* \geq 2/n$ several kinds of weak-ties will not be included into any community. By using the threshold t_s^*, our solution can well handle the incorrect membership of weak-ties problem[6]. In Ref.[10], the concept of *bridge* is proposed to distinguish strong or weak ties. A bridge is a link in a network which provides the only path between two nodes. Thus, we have the following theorem:

Theorem 1 The bridge cannot be included in any LLS with $t_s^* \geq 2/n$.

Proof Given a bridge e_{pq} being the only path between i_p and i_q, there are two cases that can include e_{pq} into at least one LLS: (1) $\{i_p, i_q\}$ belongs to a cosine pattern S; (2) i_q is a keystone of a cosine pattern S', $i_p \in S'$, likewise for i_p.

Since $N_p \bigcap N_q = \emptyset$, $supp(S) \leq supp(\{i_p, i_q\}) = 0$. So, $supp(S) = 0$, and thus S is not a frequent pattern, much less a cosine pattern. The case (1) cannot happen.

Assume $S' = \{i_p, i_r\}$ is a cosine pattern, and i_q is a keystone of S'. Since $supp(S') \geq t_s^* \geq 2/n$, there is at least two keystones of S'. Besides i_q, let i_s be the other keystone. Then, another path appears between i_p and i_q, *i.e.*, $\{i_p, i_s, i_r, i_q\}$. Thus, e_{pq} is not a bridge. The case (2) also cannot happen.

Besides bridge, the local bridge is defined to be another type of weak-tie. We say an edge is a local bridge if its two endpoints have no friends in common[11]. The span of a local bridge is the distance its endpoints would be from each other if the edge is deleted. Obviously, the span of any local bridge is greater than 2, and more larger

the span is, the weaker this local bridge is. For handling the local bridge, we have:

Theorem 2 The local bridge with span more than 3 cannot be included in any LLS with $t_s^* \geq 2/n$.

Proof As consider two cases in the proof of Theorem 1, the case (1) can be easily excluded due to the same reason.

Assume e_{pq} is a local bridge with span more than 3, $S' = \{i_p, i_r\}$ be a cosine pattern, and i_q is a keystone of S'. Since $supp(S') \geq 2$, after deleting e_{pq}, the path $\{i_p, i_s, i_r, i_q\}$ still exists of which the span is 3. This contradicts with the assumption that the span of local bridge e_{pq} is more than 3.

Theorem 2 implies that by employing the threshold t_s^* our solution can maintain the strongest weak ties, since 3 is the smallest span value of any local bridge.

3. Clustering LLSs to obtain link communities

Based on the set $\mathcal{S}$ of cosine patterns returned by CoPaMi, we construct a set of LLSs $\mathcal{H}_\mathcal{S}$ of which each element $\forall H_S \in \mathcal{H}_\mathcal{S}$ is defined in Proposition 2. Now, the arising problem is how to cluster adjacent LLSs to obtain a certain number of *crisp* link communities. To this end, we here propose to use the hypergraph model to represent $\mathcal{H}_\mathcal{S}$ and then employ the hMETIS[12] tool to divide $\mathcal{H}_\mathcal{S}$ into a certain number of clusters. In mathematics, a hypergraph is a generalized graph in which an edge can connect any number of vertices.

Definition 2 A link hypergraph is defined as $HG = \{E', \mathcal{H}_\mathcal{S}, \omega\}$. The node set E' is a subset of *edge* set of G, *i.e.*, $E' \subseteq E$. $\mathcal{H}_\mathcal{S}$ and ω are set of hyperedges and their weights, respectively. Each hyperedge $H_S \in \mathcal{H}_\mathcal{S}$ corresponds to a LLS, and its weight $\omega_S = \cos(S)$.

In our hypergraph model, every hyperedge corresponds to a LLS. We employ a top-down strategy based on the cut optimization for LLS clustering. Since hypergraph partitioning is a ripe technology, in this paper, we just use the software package called hMETIS for this task.

III. Algorithmic Process

The steps of the proposed OCD method can be summarized in Algorithm 1. Steps 1–4 have been elaborated in Section II. Here, we just give a brief introduction to the last step called membership translation. The OCD method is better to provide the membership of a node for quantifying how strongly this node belongs to a particular community. Based on the K link communities, 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. For instance, assume a node having 3 links (*i.e.*, its degree is 3), among which 1 link belongs to C_1 and 2 links belong to C_2. So, this overlapping node attaches to C_1 with 1/3 probability and C_2 with 2/3 probability.

Algorithm 1 OCD based on LLS mining and clustering

Input:	the transaction dataset $\mathcal{D}$ of G, two thresholds t_s^* and t_c^*, and the number of communities K;
Output:	The membership matrix $\boldsymbol{U}$;
1:	Invoke the 'CoPaMi' algorithm on $\mathcal{D}$ to mine the cosine pattern set w.r.t. t_s^* and t_c^*, denoted by $\mathcal{S}$;
2:	Search keystones of $\mathcal{S}$ and obtain the LLS set $\mathcal{H}_\mathcal{S}$;
3:	Construct a hypergraph HG of $\mathcal{H}_\mathcal{S}$;
4:	Invoke the 'hMETIS' tool on HG to obtain K clusters;
5:	Restore the membership of a node from link communities (denoted by $\boldsymbol{U}$), and return $\boldsymbol{U}$.

IV. Experimental Results

In this section, we report the comparison results with state-of-the-art OCD methods on both synthetic and real-world networks. Six baseline tools are selected for comparison: 1) CFinder (F) is the implementation of the k-cliques percolation method; 2) Link (L) is the link-community based method; 3) MOSES (M) is a stochastic block model based local optimization scheme; 4) GCE (G) is a local greedy optimization strategy taking k-cliques as seed; 5) SVI (S) is a recent method based on link communities; and 6) BigClam (B) is a recent global optimization model.

1. Synthetic networks

For synthetic networks we generate unweighted, undirected and overlapping networks with ground-truth communities using LFR benchmark. We set the exponent $\tau_1 = 2$ and $\tau_2 = 1$ to shape the pow law distributions of node degree and community size. The maximum degree k_{max} is set to 50 and the community size s lies between 10 and 50. Thus, we generate a synthetic network with 1000 nodes among which 10% nodes belong to multiple communities. Two parameters are tuned in our experiments, the number of communities, denoted by O_m, to which each overlapping node belongs and the mixing parameter μ respectively. Note that smaller value of μ indicates more clear community structures.

We adopt the Normalized mutual information (NMI) for overlapping networks as the performance measure on synthetic networks. Fig.1(a) shows the detailed comparison in terms of NMI as μ varies from 0.1 to 0.5 in the case of $O_m = 2$. As can be seen, the accuracy of all methods seems decline with the increasing of μ. The similar declining trend can also be observed in Fig.1(b) as O_m varies from 2 to 6 and $\mu = 0.3$. This is caused by the harder OCD task as the increase of μ and O_m. Obviously, the performance of our method is consistently better than the other six in almost all cases except for the one while μ is 0.1. This is because more highly overlapped communities can be detected by our method when μ is 0.1.

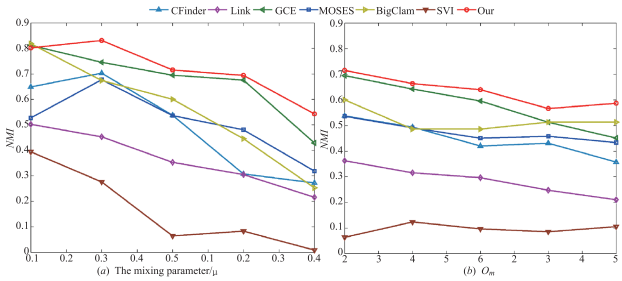

Fig. 1. Performance comparison on LFR networks

2. Real-world networks

Seven real-life social networks are used in our experiments. More information about these networks can be found in Table 1, in which $\langle k\rangle = 2|E|/|V|$ denotes the average degree and C denotes the average clustering coefficient.

Table 1. Statistics of real-world networks

Network	$\|V\|$	$\|E\|$	$\langle k\rangle$	$\langle C\rangle$
Dolphins	62	159	5.13	0.303
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
Oklahoma	17420	892524	102.47	0.23
Amazon0302	262111	1234877	9.42	0.420

We adopt overlapping modularity Q^{ov}, one of the most widely-used measures, as the validation measure:

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

where A_{pq} is 1 if there is an edge between node i_p and i_q, 0 otherwise, and $|N_p|$ indicates the number of neighbors of node i_p as well as the degree of node i_p. The value of Q^{ov} lies between –1 and 1. The larger Q^{ov} is, the better quality is.

Table 2. Performance comparison on real-world networks

Method / Network	F	L	G	M	B	S	O
Dolphins	0.41	0.11	0.46	0.43	0.41	0.25	**0.49**
LesMis	0.19	0.31	0.47	0.49	0.45	0.39	**0.50**
Google	0.10	0.12	0.06	0.23	0.27	0.21	**0.29**
Protein	0.26	0.26	0.44	0.37	**0.49**	0.24	**0.47**
Words	0.07	0.23	0.22	0.12	0.24	0.11	**0.48**
Oklahoma	0.06	0.01	0.27	0.02	0.07	0.10	**0.29**
Amazon0302	0.46	0.45	0.60	0.62	0.56	—	**0.68**

Table 2 shows the comparison results in terms of Q^{ov}. The value of SVI on Amazon0302 network is missing due to its scalability. Except for the Protein network, our method (O) consistently obtains the highest score, which implies the quality of overlapping communities discovered by our method is higher than other tools. More interestingly, the results are generally consistent with Fig. 1. That is, an evident order of five methods could be observed: "O→G→B→M→C→L→S".

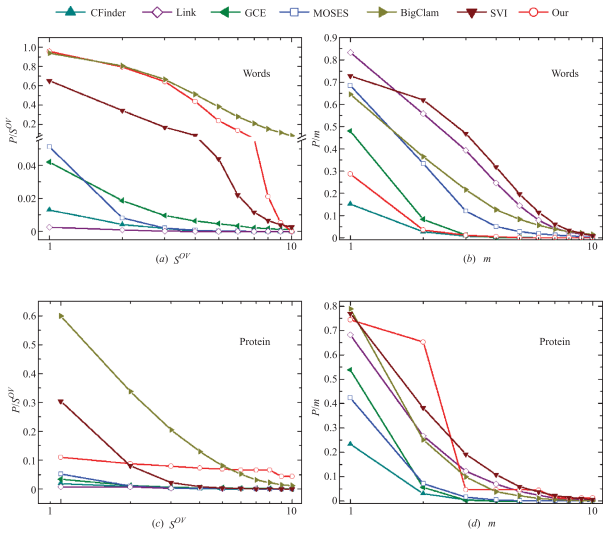

Fig. 2. The cumulative distribution of S^{ov} and m

To give a close look at identified communities, we investigate the two cumulative distribution functions denoted by $P(S^{ov})$ and $P(m)$, where S^{ov} is defined as the overlap size between two communities and m is defined as the number of communities one node belongs to. Fig.2 depicts $P(S^{ov})$ and $P(m)$ of five methods on Words and Protein networks. Overall, our method can successfully capture multiple relationships and a great deal of overlaps. One exception occured in Fig.2(b): $P(m)$ of our method is obviously low, but $P(S^{ov})$ on Words is relatively high. By observing the topological structure of Words, we find that this network contains a large number of weak-ties. So, our method rules out more nodes linked by the weak-tie and thus maintains few nodes.

V. Related Work

Recent years have witnessed a growing interest in overlapping community detection. Two far-reaching idea are k-clique percolation[2] and link partitioning[3]. Whereafter, local expansion and fuzzy detection are presented for the OCD problem[1]. The former uses close-knit structures as seeds and then expands them based on a local benefit function, such as the famous GCE method. The latter adopts appropriate models to get soft membership vector for each node. There are many models such as mixture models, non-Negative matrix factorization (NMF) and Stochastic block model (SBM). MOSES is based on SBM, yet BigClam is based on NMF. The proposed method can be viewed as a combination of local structures mining with link partitioning.

VI. Conclusion

This paper starts by extending the similarity of link pairs to get the definition of so-called LLS. We prove that LLS with an appropriate threshold can successfully filter

weak-ties. Mining and clustering LLSs are explored, and thus integrated to form a novel OCD method. Thorough experiments are conducted on a series of synthetical networks generated by varing parameters and also on seven real-life networks. Results in terms of NMI and Q validate our method has advantage over six existing methods on discovering higher quality communities.

References

[1] 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.

[2] G. Palla, I. Derenyi, I. Farkas and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society", *Nature*, Vol.435, No.7043, pp.814–818, 2005.

[3] 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.

[4] J. Zhang, K. Deng, *et al.*, "Community detection in complex networks based on link label propagation", *Chinese Journal of Electronics*, Vol.43, No.6, pp.1113–1118, 2015.

[5] L. Pan, J. Jin, *et al.*, "Detecting link communities based on local information in social networks", *Chinese Journal of Electronics*, Vol.40, No.11, pp.2255–2263, 2012.

[6] S. Lim, S. Ryu, S. Kwon, *et al.*, "LinkSCAN*: Overlapping community detection using the link-space transformation", *2014 IEEE 30th International Conference on Data Engineering (ICDE)*, pp.292–303, 2014.

[7] 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.

[8] 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.

[9] Z. Wu, J. Cao, J. Wu, *et al.*, "Detecting genuine communities from large-scale social networks: A pattern-based method", *The Computer Journal*, Vol.59, No.7, pp.1343–1357, 2014.

[10] M. Granovetter, "The strength of weak ties: A network theory revisited", *Sociological Theory*, Vol.1, No.1, pp.201–233, 1983.

[11] D. Easley, J. Kleinberg, "Networks, crowds, and markets: Reasoning about a highly connected world", *Cambridge University Press*, 2010.

[12] 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.

TAO Haicheng was born in 1987. He received the M.S. degree in software engineering from University of Science and Technology of China in 2012. He is now a Ph.D. candidate of Nanjing University of Science and Technology. His research interests include social network analysis and data mining. (Email: haicheng.tao@gmail.com)

WANG Youquan (corresponding author) was born in 1984. He received the M.S. degree in computer science from Nanjing University of Finance and Economics in 2009. He is now a Ph.D. candidate of Nanjing University of Science and Technology. His research interests include social network analysis and data mining. (Email: youq.wang@gmail.com)