A Better Strategy of Discovering Link-Pattern based Communities by Classical Clustering Methods

Chen-Yi Lin[1], Jia-Ling Koh[2], Arbee L. P. Chen[3]

[1] Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
[2] Department of Computer Science and Information Engineering,
National Taiwan Normal University, Taipei, Taiwan
[3] Department of Computer Science, National Chengchi University, Taipei, Taiwan
[1] d9562820@oz.nthu.edu.tw, [2] jlkoh@csie.ntnu.edu.tw, [3] alpchen@cs.nccu.edu.tw

Abstract. The definition of a community in social networks varies with applications. To generalize different types of communities, the concept of link-pattern based community was proposed in a previous study to group nodes into communities, where the nodes in a community have similar intra-community and inter-community interaction behaviors. In this paper, by defining centroid of a community, a distance function is provided to measure the similarity between the link pattern of a node and the centroid of a community. The problem of discovering link-pattern based communities is transformed into a data clustering problem on nodes for minimizing a given objective function. By extending the partitioning methods of cluster analysis, two algorithm named G-LPC and KM-LPC are proposed to solve the problem. The experiment results show that KM-LPC outperforms the previous work on the efficiency, the memory utilization, and the clustering result. Besides, G-LPC achieves the best result approaching the optimal solution.

Keywords: Social Network, Link-Pattern based Community, Clustering Algorithms

1 Introduction

Social network analysis is an established field in sociology. A social network is mostly modeled by a graph in which a node represents an individual and an edge between two nodes denotes a social interaction between the corresponding individuals. In recently years, because of the increasing availability of social network data on the Web 2.0 platform, the study of social network analysis has emerged into an active research field. The community structure is an important topological characteristic of social networks, which provides a basis for further analysis of social networks. Accordingly, discovering the communities from a social network has become an essential problem on social network analysis.

The definitions of a community in social networks vary with applications. In most studies, finding groups of nodes within which the interconnections are dense but between which the interconnections are sparse is attractive to users. In earlier papers, the

graph partitioning techniques were adopted to divide nodes into subsets by discovering the various kinds of cuts in a graph such as average cuts [1], normalized cuts [9], min-max cuts [2], and maximum flow/minimum cuts [3, 5]. However, in some applications such as those in blogosphere, a group of individuals linking to the same set of blogs indicates a set of latent friends with common interests even though they sparsely link to each other [6, 8]. Therefore, to generalize the different types of communities, the concept of *link-pattern based community* was proposed in [7].

A link-pattern based community is a group of nodes which have a similar link patterns, i.e., the nodes in the same community have similar intra-community and inter-community interaction behaviors. For example, the individuals, denoted by the nodes in Figure 1, are grouped into three communities: $C_1 = \{v_1, v_2, v_3, v_4\}$, $C_2 = \{v_5, v_6, v_7, v_8\}$, and $C_3 = \{v_9, v_{10}, v_{11}, v_{12}\}$. The nodes in C_1 link densely to each other, link moderately to the nodes in C_2, and link sparsely to the nodes in C_3. On the other hand, the nodes in C_2 link moderately to the nodes in C_1 and link sparsely to the nodes in both C_2 and C_3. The nodes in C_3 also link to other nodes within the community and between the communities in similar ways. The concept of a *community prototype graph* was also proposed, which consists of a set of *community nodes* (i.e., C_1, C_2 and C_3 in Figure 1) and a set of edges among community nodes to represent the community structures. Accordingly, the graph and its community prototype graph are represented as affinity matrices in which each entry represents the weight of an edge between two corresponding nodes. An iterative algorithm named CLGA was developed to find the optimal community prototype graph from the graph by solving the optimization problem of matrix approximation.

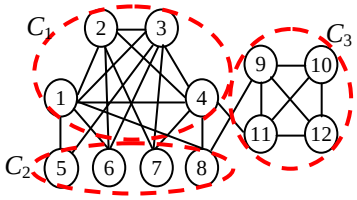

Figure 1. An example of the link-pattern based communities

The number of individuals in a social network is enormous in most cases, and the size of the affinity matrix of the original graph is determined by the number of individuals. Consequently, the algorithms for solving the problem of discovering the link-pattern based communities are challenging on the requirement of memory usage and the performance efficiency. However, in [7], in order to find the optimal community prototype graph, it requires exhaustive search by moving a node from one community to another community. In each time of iteration, the affinity matrix of the corresponding community prototype graph has to be recomputed. As a result, CLGA is computationally infeasible. Besides, CLGA has to maintain the affinity matrix of the community prototype graph. Consequently, the memory requirement of CLGA is at least double of the one required by the affinity matrix of the original graph.

According to the concept of the link-pattern based community, the edges with weights incident to a node are essential features which imply the link-pattern of the node. In this paper, we reformulate the problem based on the proximity of the links of nodes to discover the link-pattern based community structures, and evaluate the quali-

ty of the community structures according to the similarity of the weights of the intra-links within a community and that of the weights of the inter-links between the community and every other community. It is proved that the reformulated problem of communities discovering is equivalent to the problem defined in [7]. In order to get a good clustering result, two different strategies are provided to select sample nodes for determining the initial the communities. Based on the extracted initial community structures, two algorithms, named G-LPC and KM-LPC, based on the classical clustering methods, are provided to discover the communities. The experiment results show that KM-LPC outperforms CLGA not only on the efficiency and memory utilization, but also on the clustering result. Although G-LPC requires the most computational cost than the others, it achieves the best result approaching the optimal solution.

The remaining sections of this paper are organized as follows. The reformulated problem and the proposed algorithms are described in Sections 2 and 3, respectively. The performance study is reported in Section 4, which shows the effectiveness, efficiency, and memory usage of the proposed methods. Finally, in Section 5, we conclude this paper and discuss directions for our future studies.

2 Preliminaries

In this section, the problem proposed in [7] for discovering the link-pattern based communities is introduced briefly. Then we reformulate the problem and provide solutions to the problem in Section 3.

2.1 Problem of Matrix Approximation Optimization

Suppose an undirected weighted graph $G = (V, E, A)$ is given, where V is a set of nodes $\{v_1, v_2, \ldots, v_n\}$, E is a set of edges (v_i, v_j), and A is the affinity matrix of G. A is an $n \times n$ symmetric matrix, in which $A[i, j]$ is a positive value representing the weight of the edge between nodes v_i and v_j if $(v_i, v_j) \in E$; otherwise, $A[i, j]$ is set to be 0.

A community prototype graph defined in [7] consists of a set of community nodes and a set of edges among community nodes associated with weights to represent the community structures. Let K denote the number of communities specified by the users. The *community structure matrix* B is a $K \times K$ matrix for representing the weights of intra-links and inter-links of the community nodes. Besides, an $n \times K$ matrix C with binary values denotes the community membership of each node, where each node belongs to exact one community and there is no empty community. The affinity matrix of a community prototype graph, denoted as A', is an $n \times n$ matrix which is the result of CBC^T. Accordingly, the challenge of discovering the link-pattern based communities is how to find C and B such that $\|A - A'\|^2$ is minimized.

[Example 2.1] The nodes of the social network shown in Figure 2(a) are required to be grouped into two link-pattern based communities. The affinity matrix of the corresponding graph is shown as Figure 2(b). When the two communities are constructed as $C_1 = \{v_1, v_2, v_3, v_4\}$ and $C_2 = \{v_5, v_6, v_7, v_8\}$, the corresponding community prototype graph shown in Figure 2(c) is the optimal solution of this case. Accordingly, the

corresponding matrices *C* and *B* of the constructed community structures are shown in Figure 2(d). Besides, the affinity matrix of the community prototype graph is shown as Figure 2(e). Therefore, the difference between the affinity matrix of the graph and the affinity matrix of the community prototype graph is 3.5.

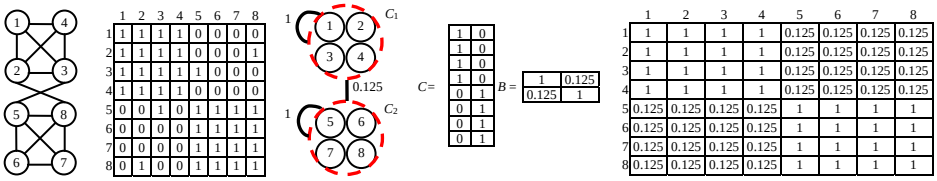

Figure 2. An example of a graph and its optimal community prototype graph

2.2 Problem Definition

Based on the definition of the link-pattern based community, a good solution of the problem tends to group the nodes with similar intra-community and inter-community interaction behaviors into the same community. The link pattern of a node is characterized by its edges linked to other nodes, and the link pattern of a community by the aggregate link patterns of the nodes in the community. An object function is designed to evaluate the quality of the communities by the distance between the link patterns of each community and its nodes.

Suppose the members in each community have been assigned. Let $C_u = \{v_{u_1}, v_{u_2}, \ldots, v_{u_{n_u}}\}$ and $C_v = \{v_{v_1}, v_{v_2}, \ldots, v_{v_{n_v}}\}$ denote two communities, where n_u and n_v denote the number of nodes in C_u and C_v, respectively. The affinity matrix for the nodes in C_u, denoted A_{C_u}, is an $n_u \times n_u$ sub-matrix of A which is the affinity matrix of the original graph. The intra-community link pattern of v_{u_i} is represented by the i^{th} row in A_{C_u} with the weights of the intra-community edges of v_{u_i}. Moreover, the affinity matrix A_{C_u, C_v} for the nodes in C_u with the nodes in C_v is an $n_u \times n_v$ sub-matrix of A, in which the i^{th} row represents the inter-community edges of v_{u_i} with the nodes in C_v.

Consequently, the intra-community pattern of the community C_u is represented by a vector of n_u dimensions where each dimension contains the average weight of the intra-community link patterns of all the nodes in C_u as the following formula shows:

$$AVG_{C_u} = \frac{\sum_{i=1}^{n_u}\sum_{j=1}^{n_u} A_{C_u}[i,j]}{n_u \times n_u} \tag{1}$$

The inter-community pattern of the community C_u with C_v is represented by a vector of n_v dimensions where each dimension contains the average weight of the inter-community edges of all the nodes in C_u with C_v as the following formula shows:

$$AVG_{C_u, C_v} = \frac{\sum_{i=1}^{n_u}\sum_{j=1}^{n_v} A_{C_u, C_v}[i,j]}{n_u \times n_v} \tag{2}$$

Therefore, the *intra-distance* of C_u and the *inter-distance* between C_u and C_v are defined by the following formulas:

$$SSD_{C_u} = \sum_{i=1}^{n_u}\sum_{j=1}^{n_u}\left(A_{C_u}[i,j] - AVG_{C_u}\right)^2 \tag{3}$$

$$SSD_{C_u,C_v} = \sum_{i=1}^{n_u}\sum_{j=1}^{n_v}\left(A_{C_u,C_v}[i,j] - AVG_{C_u,C_v}\right)^2 \tag{4}$$

[Example 2.2] The nodes of the graph shown in Figure 2(a) are grouped into C_1 and C_2. Consequently, the corresponding matrices A_{C_1}, A_{C_2}, A_{C_1,C_2}, and A_{C_2,C_1} are shown in Figure 3(a), (b), (c), and (d), respectively. Accordingly, the values of AVG_{C_1}, AVG_{C_2}, AVG_{C_1,C_2}, and AVG_{C_2,C_1} are 1, 1, 0.125, and 0.125, respectively. Besides, the values of SSD_{C_1}, SSD_{C_2}, SSD_{C_1,C_2}, and SSD_{C_2,C_1} are 0, 0, 1.75, and 1.75, respectively.

	1	2	3	4
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1

(a)

	5	6	7	8
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1

(b)

	5	6	7	8
1	0	0	0	0
2	0	0	0	1
3	1	0	0	0
4	0	0	0	0

(c)

	1	2	3	4
5	0	0	1	0
6	0	0	0	0
7	0	0	0	0
8	0	1	0	0

(d)

Figure 3. The affinity matrices A_{C_1}, A_{C_2}, A_{C_1,C_2}, and A_{C_2,C_1}

Consequently, the problem of discovering the link-pattern based communities from a social network is formulated by minimizing the sum of the intra- and inter-distances of the communities as the following. According to a given positive integer K which denotes the number of communities, the optimized communities are given by minimizing the following objective function:

$$\arg \min_{C_1,C_2,\ldots,C_K} \sum_{i=1}^{K}\left(SSD_{C_i} + \sum_{j=1\wedge j\neq i}^{K} SSD_{C_i,C_j}\right) \tag{5}$$

where the n nodes in the social network are separated into C_1, …, C_K such that each node belongs to exact one community. Besides, there is no empty community allowed.

It is deducible that the optimal solution discovered according to the proposed objective function is the same as the one discovered by CLGA. Suppose the members in each community have been assigned. Based on our observation on matrix B, the diagonal entry $B[i, i]$ is equal to AVG_{C_i} and the non-diagonal entry $B[i, j]$ in B is equal to AVG_{C_i,C_j}. Therefore, for any pair of v_x and v_y in C_i, the entry $A'[x, y]$ in A' of the community prototype graph is equal to AVG_{C_i}. On the other hand, the entry $A'[x, y]$ is equal to AVG_{C_i,C_j} for any node v_x in C_i and node v_y in C_j. Since each node belongs to exact one community, $||A - A'||^2$ is equal to $\sum_{i=1}^{K}\left(SSD_{C_i} + \sum_{j=1\wedge j\neq i}^{K} SSD_{C_i,C_j}\right)$.

3 The Proposed Algorithms

According to the objective function defined in Section 2, the task of discovering the link-pattern based communities is an optimization problem of minimizing the objective function. However, it is computationally infeasible to exhaustively search the global optimum solution of this problem. To provide a heuristic algorithm for solving

this problem, a greedy based algorithm and a K-Means based algorithm are proposed to discover the disjoint clusters of data nodes which correspond to the link-pattern based communities of these nodes.

3.1 Basic Idea

In order to get a value of the objective function as small as possible, one effective way is to adopt the greedy-based algorithm in which each node is iteratively assigned to a community such that the obtained value of the objective function is minimal. The K-Means algorithm [4] is a typical method of cluster analysis. The goal of K-Means is to minimize the sum of squared distances between data and the mean of the corresponding cluster, which is similar to the goal of the objective function defined in this paper. Therefore, a K-Means based algorithm is also proposed to discover the communities.

According to the given information of graph G, row i in the affinity matrix A of G provides the information of the link pattern of node v_i, which forms the feature vector of v_i. For a community C_u, the feature vector of C_u is an n-dimensional vector. The j^{th} dimension in the feature vector of C_u contains the value of AVG_{C_u} if node v_j belongs to C_u; otherwise, it contains the value of AVG_{C_u,C_v} if node v_j belongs to another community C_v. During the progressive process of clustering, the feature vector of a community will be used to be its centroid. Therefore, the value of the objective function defined in formula (5) corresponds to the sum of squared distance between the feature vector of each node and the centroid of its community.

To adopt the clustering methods for discovering the communities, first, K initial centroids are determined by performing a clustering on the sample nodes selected from the social network. Next, each node in the social network is assigned to a community by executing one of the proposed two algorithms. The centroid of a community will be updated according to the nodes assigned to the community. Relative to the new centroids, the above process is repeated until there is no change in communities.

3.2 Determining Initial Centroids

In terms of the quality of the clustering result, determining a set of appropriate initial centroids of clusters is the key step of clustering algorithms. However, it is not easy to determine a 'good' set of initial centroids of clusters without knowing the connectivity among the nodes. Therefore, some sample nodes are chosen from the graph. Then the agglomerative hierarchical clustering method is adopted to separate these sample nodes into K disjoint clusters. Finally, the mean of the feature vectors of the sample nodes assigned to a cluster is set as the initial centroid of the cluster.

A straightforward method is to choose the sample nodes randomly. However, a good quality of the clustering result is obtained by chance based on which sample nodes are chosen. In order to understand the characteristics of the link-pattern based communities in a real dataset, the Enron email dataset is analyzed by CLGA. It is observed that the nodes which are distributed to the same community usually have the similar degrees. In other words, it is more possible that two nodes belong to the same community as their degrees are closer. Accordingly, in our second strategy, the sam-

ple nodes are selected based on the degrees of the nodes. Moreover, the number of chosen sample nodes is determined by $K \times U$, where K is the number of expected communities and U is a user-specified integer which is at least one and less than $\lfloor n/K \rfloor$.

By summarizing the above considerations, the following two strategies are used to select sample nodes from the given graph G.

(1) Picking by random

$K \times U$ sample nodes are chosen from G randomly without replacement.

(2) Picking by node degree

The nodes with the identical degree are assigned to the same group. Within each group, U nodes are chosen randomly without replacement.

After selecting sample nodes by one of the above-mentioned strategies, the sample nodes are separated into K clusters by the agglomerative hierarchical clustering. At the beginning, each sample node is considered as an individual cluster. Let c_x and c_y denote the centroids of two clusters C_x and C_y, respectively. The distance between C_x and C_y is decided by calculating the Euclidean distance between c_x and c_y:

$$dist(C_x, C_y) = \|c_x - c_y\| \tag{6}$$

Iteratively, two nearest clusters are chosen to be merged into a cluster until K clusters remain. Whenever two clusters are merged to generate a new cluster C_l, the centroid of C_l, which is denoted as c_l, is obtained according to the following formula:

$$c_l = \frac{1}{n_l} \sum_{SN \in C_l} SN.fv \tag{7}$$

where n_l denotes the number of sample nodes in C_l, SN denotes a sample node in C_l, and $SN.fv$ denotes the feature vector of the sample node.

[Example 3.1] In the graph shown in Figure 2(a), the number of distinct degree values of nodes is 2. By using the picking by node degree strategy to select sample nodes, the nodes in the graph are separated into two groups $\{v_1, v_4, v_6, v_7\}$ and $\{v_2, v_3, v_5, v_8\}$. When U is set as 1, the number of chosen sample nodes from each group is 1. Suppose nodes v_4 and v_3 are chosen as the sample nodes. Accordingly, <1, 1, 1, 1, 0, 0, 0, 0> and <1, 1, 1, 1, 1, 0, 0, 0> of v_4 and v_3 are used as the initial centroids of C_1 and C_2.

3.3 Communities Discovering

After the initial centroids of K clusters are obtained, each node in G is then assigned to the closest cluster, and the centroid of each cluster is updated according to the nodes assigned to the cluster. Then two algorithms are proposed to reassign each node in G to the clusters iteratively until the result converges.

(1) Discovering Initial Communities

According to the initial centroids, each node in the graph G is assigned to the closest cluster one by one. The distance between a node v and C_u is determined:

$$dist(v, C_u) = \|v.fv - c_u\| \tag{8}$$

where $v.fv$ denotes the feature vector of node v.

When the assignment of all the nodes to the clusters completes, the initial structures of communities are constructed. Accordingly, the feature vector of each initial community is computed to update its centroid.

[Example 3.2] By continuing the result of Example 3.1, the nodes in the graph are separated into two initial clusters C_1 = {v_1, v_2, v_4} and C_2 = {v_3, v_5, v_6, v_7, v_8}. The values of AVG_{C_1}, AVG_{C_2}, and AVG_{C_1,C_2} are 1, 0.76, and 0.2667, respectively. Therefore, the centroids of C_1 and C_2 are <1, 1, 0.2667, 1, 0.2667, 0.2667, 0.2667, 0.2667> and <0.2667, 0.2667, 0.76, 0.2667, 0.76, 0.76, 0.76, 0.76>, respectively.

(2) The Clustering Algorithms

The two clustering algorithms proposed to discover the community structures according to the initial communities are introduced.

(*A*) *The Greedy based Algorithm*

The process of the Greedy based algorithm for discovering Link Pattern-based Communities (abbreviated as G-LPC) aims to distribute a node to the cluster such that the objective function is minimized locally. In each time of iteration, one by one, each node is checked to decide the cluster which the node is assigned to. The node is moved from the cluster, which it was assigned to in last time of iteration, to another cluster if the value of objective function will be reduced after the movement. The above process is repeated until there is no change in clusters.

[Example 3.3] According to the result of Example 3.2, the two initial clusters are C_1 = {v_1, v_2, v_4} and C_2 = {v_3, v_5, v_6, v_7, v_8}. If we move node v_1 from C_1 to C_2, AVG_{C_1}, AVG_{C_2}, and AVG_{C_1,C_2} are recomputed to be 1, 0.6111, and 0.4167, respectively. Also, the centroids of C_1 and C_2 are updated. Finally, the new value of the objective function is obtained, which is 14.3889. The value is larger than the previous one, i.e. 10.4267; hence, v_1 is remained in C_1.

(*B*) *The K-Means based Algorithm*

The process of the K-Means based algorithm for discovering Link Pattern-based Communities (abbreviated as KM-LPC) assigns each node to the cluster whose centroid is nearest to the feature vector of the node. At the end of each time of iteration, the centroid of a cluster is recomputed according to the nodes which are assigned to the cluster. The above process is repeated until no member of any cluster changes.

[Example 3.4] By continuing the result of Example 3.2, the initial centroids of C_1 and C_2 are <1, 1, 0.2667, 1, 0.2667, 0.2667, 0.2667, 0.2667> and <0.2667, 0.2667, 0.76, 0.2667, 0.76, 0.76, 0.76, 0.76>, respectively. The distances between the feature vector <1, 1, 1, 1, 0, 0, 0, 0> of v_1 and the centroids of C_1 and C_2 are 0.9068 and 1.9953; consequently, v_1 is remained in C_1. Similarly, other nodes are assigned to the closest clusters. In the end of this loop, the members of C_1 and C_2 are {v_1, v_2, v_3, v_4} and {v_5, v_6, v_7, v_8}. By KM-LPC, the final community result is C_1 = {v_1, v_2, v_3, v_4} and C_2 = {v_5, v_6, v_7, v_8}, which is the optimal solution of the link-pattern based communities with 2 communities.

4 Performance Evaluation

In order to evaluate the effectiveness, efficiency, and memory requirement of the proposed algorithms, G-LPC and KM-LPC are implemented by MATLAB ver. 7.0.1. Furthermore, CLGA [7] is also implemented for comparison. All the experiments are performed on a personal computer with the Intel Pentium Core 2 Quad CPU, 2 GB of main memory, and running the Microsoft Windows XP.

4.1 Datasets

Two real datasets: Enron email DB[1] and DBLP Bibliography DB[2] are used in the following experiments. The Enron email DB contains the emails of 151 employees. The dataset is modeled by a graph in which the weight of the edge between two nodes is set to be 1 if any one of the corresponding employees has ever sent an email to the other; otherwise, the edge weight is set to be 0. On the other hand, the DBLP dataset contains the publication information of approximate 700,000 authors. In order to reduce the size of the dataset, we select the authors who have at least 75 coauthors, and the authors who publish more than 10 papers with one of the previously selected authors to run the experiments. As a result, only 7,356 authors are selected. Then an undirected weighted graph is constructed, in which a node represents one of the selected 7,356 authors; in addition, the weight of an edge between two nodes is set to be the number of collaborations between the two corresponding authors normalized by the maximum number of collaborations between any two authors.

4.2 Results and Discussions

There are three parts of experiments to be performed. In the first part, the Enron email dataset is used to evaluate the quality of obtained communities, the execution time, and the memory requirement of the proposed algorithms and the previous work. Next, by using the DBLP dataset, the detailed comparisons of the parameters setting in KM-LPC are observed in the second part of experiments. At last, the properties of the discovered link pattern-based communities from the DBLP dataset are analyzed.

4.2.1 Comparison between the Proposed Algorithms and CLGA

[Exp. 1] The Enron email dataset is used in this part of experiments. The parameter U is set to be 1. In addition, the strategy of picking by node degree is adopted.

By varying the value of K, the values of the objective function for the communities discovered by the proposed two algorithms and CLGA are shown in Figure 4(a). It is indicated that the community structures obtained by the two proposed algorithms are both better than the one obtained by CLGA. Besides, G-LPC gets the best result. In CLGA, if users have no prior knowledge about the social network, the initial setting of the community structures is determined by random. The result shows that the random setting adopted in CLGA usually results in a poorer result by comparing with our algorithms. Figure 4(b) shows the execution time of the algorithms, in which the execution time of KM-LPC is much less than the time of the others. It shows that KM-LPC provide a significant improvement for discovering the communities efficiently. Moreover, the sizes of memory requirement of the algorithms are shown in Figure 4(c). In our algorithms, in addition to the affinity matrix of the given graph, only the centroids of the communities have to be maintained in main memory instead of storing another affinity matrix of the community prototype graph adopted by CLGA. Therefore, both the proposed algorithms require less memory than CLGA.

[1] http://www.cs.cmu.edu/~enron/
[2] http://www.informatik.uni-trier.de/~ley/db/

To decide the community of a node, G-LPC computes the new value of the objective function for each possible movement of the node. On the other hand, in KM-LPC, it only computes the distance between the feature vector of a node and all the centroids of communities to find the community with the nearest centroid. As a result, although G-LPC requires the most computational cost than the others, it achieves the better result than the others when the value of *K* is increasing. Besides, when the strategy of picking by random is adopted, the performances of our algorithms are also better than that of CLGA. Due to space restrictions, the details of this part of experiment are not shown here.

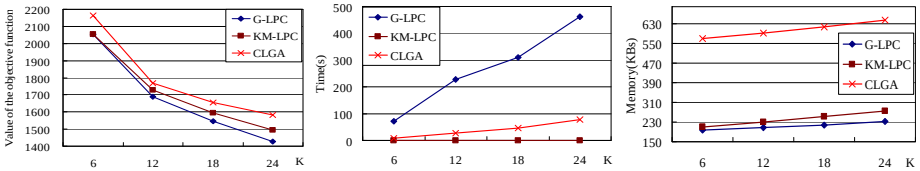

Figure 4. Comparisons between our algorithms and CLGA in the Enron email dataset

4.2.2 Comparison of the Parameters Setting in KM-LPC

Because the number of authors chosen from the DBLP dataset is 7,356, the graph constructed for the dataset is large and complicated. CLGA cannot run on the DBLP dataset under limited memory. Thus, in this part of experiments, we will observe the effect of the parameters *U* and *K* on KM-LPC. Two versions of the algorithm are implemented where the strategy for selecting the sample nodes adopts the picking by random and picking by node degree, individually.

[Exp. 2] The parameter *U* is set to be 1 and the value of *K* is varied from 100 to 220. The results of the objective function of the 2 different versions of KM-LPC are shown in Table 1. The bold-faced values shown in the table indicate the better result in the two versions of KM-LPC. In most cases, it obtains the better result of the communities by using the picking by node degree strategy than using the picking by random strategy. However, since most of the edge weights in the graph are very small, the difference between the obtained objective function values of these two strategies is not obvious. Table 2 shows the execution time of the 2 versions of KM-LPC, in which the bold-faced values represent the same meaning as used in Table 1. According to the results, for KM-LPC, the version by adopting the picking by node degree strategy runs faster than the one by adopting the picking by random strategy in most cases.

[Exp. 3] For a setting of *K*, the value of *U* is varied from 1 to 3. In addition, KM-LPC is performed by combined with the picking by node degree strategy. Table 3 shows that a larger value of *U* gets smaller value of the objective function in most cases. That is, for KM-LPC, picking more sample nodes from the graph to determine the initial centroids of the clusters is a good strategy to get better result. However, when the value of *U* increases, the number of selected sample nodes increases. The cost of performing the hierarchical clustering to determine the initial centroids of clusters also increases tremendously. Therefore, as the results shown in Figure 5, the execution time of KM-LPC substantially increases when the value of *U* increases.

Table 1. The values of the objective function obtained by Exp. 2

Algo. \ K	100	115	130	145	160	175	190	205	220
KM-LPC (Random)	7212.5	7207.7	**7186.1**	7178.1	7162.6	7149.0	7132.1	7119.4	7101.3
KM-LPC (Degree)	**7211.0**	**7206.8**	7191.6	**7177.1**	**7154.8**	**7144.1**	**7130.2**	**7118.5**	**7098.9**

Table 2. The running time (sec.) obtained by Exp. 2

Algo. \ K	100	115	130	145	160	175	190	205	220
KM-LPC (Random)	915.1	653.5	935.5	1320.2	1396.1	1459.5	1576	**1182.4**	**1431**
KM-LPC (Degree)	**518.3**	**643.6**	**743.8**	**811.4**	**879.1**	**1048**	**1152.2**	1258.7	1604.8

Table 3. The values of the objective function by varying the values of K and U

U \ K	100	115	130	145	160	175	190	205	220
1	7211.0	7206.8	7191.6	7177.1	**7154.8**	7144.1	7130.2	7118.5	7098.9
2	7210.9	7205.5	**7189.6**	7174.6	7159.7	7143.7	7130.1	7115.7	7099.0
3	**7210.7**	**7205.3**	7190.3	**7173.4**	7159.3	**7143.4**	**7127.8**	**7112.1**	**7097.2**

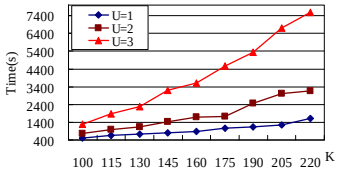

Figure 5. The running time of Exp. 3

4.2.3 Property Study of the Discovered Link Pattern-based Communities

[Exp. 4] The parameter U is set to be 1 and the value of K is varied from 25 to 200. In addition, KM-LPC is performed by combined with the picking by node degree strategy. Suppose a node v_x belongs to a community C_i. The *intra-community-interaction* of node v_x is defined to be the sum of the weights of the edges between v_x and all the other nodes belonging to C_i divided by the sum of the weights of all edges of v_x. A node with a high value of the intra-community-interaction implies that the edges of the node mainly connect to the other nodes within the same community. In this experiment, the property of the discovered communities is studied by measuring the average intra-community-interaction of the nodes in the graph. When the values of K are set to be 200, 100, 50, and 25, the values of the obtained average intra-community-interaction are 0.70, 0.79, 0.84, and 0.91, respectively. It is indicated that the value of the average intra-community-interaction tends upwards by decreasing the value of K.

By analyzing the discovered communities by KM-LPC in detail, when a small value of K is given, most of the communities have dense connections within the community and sparse connections with other communities. However, when the value of K becomes as large as 200, two different types of communities are observed. The first type of communities has dense connections within the community. Although the second type of communities has sparse connections within the community, the nodes in each community consistently connect to the nodes in a certain set of dense communities.

According to the semantics of the dataset, the authors assigned to one of the first type of communities have strong co-author relationship within the community. Thus, the members of a community in first type have common research topics. On the other hand, in the second type of communities, although the authors cooperate seldom with each other, they co-work with the authors in the same set of the first type communities. Therefore, the authors assigned to a second type community also have the similar research interests, who are potential partners with each other. This interesting finding is useful for authors to find possible cooperators. Consequently, depending on the

needs of users, KM-LPC can discover the different meaningful communities by varying the value of *K*.

5 Conclusions and Future work

In this paper, we reformulate the problem of discovering link-pattern based communities from a social network based on the similarity of link patterns of the nodes within each community. The problem of discovering link-pattern based communities is transformed to a classical clustering problem. Two algorithms named G-LPC and KM-LPC are proposed based on the classical clustering methods. The experiment results with the real datasets demonstrate that KM-LPC is better than CLGA not only on the discovered communities but also on the efficiency and memory utilization. Although the computational cost of G-LPC is higher than the others, its result is the best approaching the optimal solution. Finally, in most cases, picking by node degree is a good strategy to select the sample nodes for deciding the initial community centroids.

In some social networks, it is allowed that an individual belongs to multiple communities. To extend the concept of link-pattern based communities in this environment for identifying the communities is under our investigation. Moreover, how to determine a proper number of communities for discovering a set of semantically meaningful communities is another important issue for our future study.

References

1. P. K. Chan, M. D. F. Schlag, and J. Y. Zien, "Spectral K-Way Ratio-Cut Partitioning and Clustering," Proceedings of the 30th Design Automation Conference, pp. 749-754, 1993.
2. C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A Min-Max Cut Algorithm for Graph Partitioning and Data Clustering," Proceedings of the 1st IEEE International Conference on Data Mining, pp. 107-114, 2001.
3. G. Flake, S. Lawrence, and C. Giles, "Efficient Identification of Web Communities," Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 150-160, 2000.
4. J. A. Hartigan, "Clustering algorithms," New York: John Wiley & Sons, 1975.
5. H. Ino, M. Kudo, and A. Nakamura, "Partitioning of Web Graphs by Community Topology," Proceedings of the 14th International Conference on World Wide Web, pp. 661-669, 2005.
6. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, "Trawling the Web for Emerging Cyber-Communities," Journal of Computer Networks, Vol. 31, No. 11-16, pp. 1481-1493, 1999.
7. B. Long, X. Wu, Z. M. Zhang, and P. S. Yu, "Community Learning by Graph Approximation," Proceedings of the 7th IEEE International Conference on Data Mining, pp. 232-241, 2007.
8. P. Reddy and M. Kitsuregawa, "Inferring Web Communities through Relaxed Cocitation and Dense Bipartite Graphs," Proceedings of the 2nd International Conference on Web Information Systems Engineering, pp. 301-310, 2001.
9. J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, Is. 8, pp. 888-905, 2000.