Report No. UIUCDCS-R-2005-2538 **UILU-ENG-2005-1731**

Mining Hidden Community in Heterogeneous Social Networks

by

Deng Cai, Zheng Shao, Xiaofei He, Xifeng Yan, and Jiawei Han

March 2005

Mining Hidden Community in Heterogeneous Social Networks*

Deng Cai† Zheng Shao† Xiaofei He‡ Xifeng Yan† Jiawei Han†

† Department of Computer Science, University of Illinois at Urbana-Champaign

‡ Department of Computer Science, University of Chicago

Abstract

Social network analysis has attracted much attention in recent years. Community mining is one of the major directions in social network analysis. Most of the existing methods on community mining assume that there is only one kind of relation in the network, and moreover, the mining results are independent of the users' needs or preferences. However, in reality, there exist multiple, heterogeneous social networks, each representing a particular kind of relationship, and each kind of relationship may play a distinct role in a particular task. Thus mining networks by assuming only one kind of relation may miss a lot of valuable hidden community information and may not be adaptable to the diverse information needs from different users.

In this paper, we systematically analyze the problem of mining hidden communities on heterogeneous social networks. Based on the observation that different relations have different importance with respect to a certain query, we propose a new method for learning an optimal linear combination of these relations which can best meet the user's expectation. With the obtained relation, better performance can be achieved for community mining. Our approach to social network analysis and community mining represents a major shift in methodology from the traditional one, a shift from single-network, user-independent analysis to multi-network, user-dependant, and query-based analysis. Experimental results on Iris data set and DBLP data set demonstrate the effectiveness of our method.

Keywords

Relation Extraction, Community Mining, Multi-relational Social Network Analysis, Graph Mining

* The work was supported in part by the U.S. National Science Foundation NSF IIS-02-09199/IIS-03-08215. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

1 Introduction

With the fast growing Internet and the World Wide Web, Web communities and Web-based social networks are flourishing, and more and more research efforts have been put on Social Network Analysis (SNA) [26][35]. A social network is modeled by a graph, where the nodes represent individuals, and an edge between nodes indicates that a direct relationship between the individuals. Some typical problems in SNA include discovering groups of individuals sharing the same properties [31] and evaluating the importance of individuals [20][11]. In a typical social network, there always exist various relationships between individuals, such as friendships, business relationships, and common interest relationships.

The web itself is a huge social network. To model such a network, a lot of work on link analysis has been proposed [28][22][6][18][5]. A web page is represented as a node, and two web pages are related to each other if there is a hyperlink between them. Once the web graph is constructed, link analysis and graph partitioning algorithms can be applied to identify the communities [16][13][33].

Most of the existing algorithms on social network analysis assume that there is only one single social network, representing a relatively homogenous relationship (such as Web page linkage). In real social networks, there always exist various kinds of relations. Each relation can be treated as a **relation network**. Such kind of social network can be called *multi-relational social network* or *heterogeneous social network*, and in this paper the two terms will be used interchangeably depending on the context. These relations play different roles in different tasks. To find a community with certain properties, we first need to identify which relation plays an important role in such a community. Moreover, such relation might not exist explicitly, we might need to first discover such a hidden relation before finding the community on such a relation network.

Let us consider a simple example. In a typical human community, there may exist many relations: some people work at the same place; some share the same interests; some go to the same hospital, *etc.* Mathematically, this community can be characterized by a big graph in which the nodes represent people, and the edges evaluate their relation strength. Since there are different kinds of relations, the edges of this graph should be heterogeneous. For some tasks, we can also model this community using several homogeneous graphs. Each graph reflects one kind of relation. Suppose an infectious disease breaks out, and the government tries to find those most likely infected. Obviously, the existing relationships among people cannot play an equivalent role. It seems reasonable to assume that under such a situation the relation "*working at the same place*" or "*living together*" should play a critical role. The question becomes: "*How to select a relation that is most relevant to the disease spreading? Does there exist a hidden relation (based on the explicit relations) that best reveals the spread path of the disease?*"

These questions can be modeled mathematically as relation selection and extraction in *multi-relational* social network analysis. The problem of relation extraction can be simply stated as follows: *In a heterogeneous social network, based on some labeled examples (e.g., provided by a user*

as queries), how to evaluate the importance of different relations? Also, how to get a combination of the existing relations which can best match the relation of labeled examples? In this paper, we propose an algorithm for relation extraction and selection. The basic idea of our algorithm is to *model this problem as an optimization problem*. Specifically, we characterize each relation by a graph with a *weight matrix*. Each element in the matrix reflects the relation strength between the two corresponding objects. Our algorithm aims at finding a linear combination of these weight matrices that can best approximate the weight matrix associated with the labeled examples. The obtained combination can better meet user's desire. Consequently, it leads to better performance on community mining.

It would be interesting to relate the relation extraction problem to the *feature extraction* problem [12] in machine learning. Feature extraction aims at the discovery of the intrinsic structure of a data set. This is similar to relation extraction, but used in different scenarios. Feature extraction is used when the objects have explicit vector representation, while relation extraction is used when only relationships between objects are available. Although feature extraction has been well studied in machine learning, there is still little research effort on relation extraction so far.

The rest of this paper is organized as follows. Section 2 presents background knowledge for relation extraction. Our algorithm for relation extraction is introduced and discussed in Section 3. Two community mining algorithms are discussed in Section 4. The experimental results on the Iris data set and the DBLP data set are presented in Section 5. We discussed some potential applications of relation extraction in Section 6. Our problem solving philosophy is discussed in Section 7. Finally, we provide some concluding remarks and suggestions for future work in Section 8.

2 Background

Social network analysis

Social network analysis (SNA) is the mapping and measuring of relationships and flows between people, groups, organizations, computers, or other information/knowledge processing objects. Social network analysis as a theme has been studied for years. The classic paper of Milgram [26] might be one of the first works on SNA. It estimates that every person in the world is only six "edges" away from every other. It sets the stage for investigations into social networks and algorithmic aspects of social networks. Many recent efforts try to leverage social networks for diverse purposes, such as expertise location [20][21], mining the network value of customers [11], and discovering shared interests [31].

Previous work in sociology and statistics has suffered from the lack of data and focused on very small networks, typically in the tens of individuals [35]. With the web growing, much potential social network data are available and a lot research efforts have been put on dealing with such data.

Schwartz and Wood mined social relationships from email logs [31]. The ReferralWeb project [21] is proposed to mine a social network from a wide variety of web data, and use it to help individuals find experts who could answer their questions. Adamic and Adar tried to discover the social interactions between people from the information on their homepages [1]. Agrawal et al. analyzed the social behavior of the people on the newsgroups [2]. Moreover, the web itself can be actually viewed as a large social network. The well-known link analysis algorithms, such as Google's PageRank [28] and Kleigberg's HITS algorithm [22], can be seen as social network analysis on the web.

Community mining

With the growth of the web, community mining has attracted increasing attention. A lot of work has been done at mining the implicit communities of web pages [16][24][7] [13][33][14], scientific literature from the Web [37], and document citation database [30].

In principle, a community can be simply defined as a group of objects sharing some common properties. Community mining has many similar properties to the graph-cut problem. Kumar et al. used the bipartite graph concept to find the core of the community, and then expanded the core to get the desired community [24]. Flake et al. applied the maximum-flow and minimum-cut framework on the community mining [13]. The authority-and-hub idea [22] was also used in the community mining [16][23][8] and has several extensions [9]. The idea of frequent itemset in association rule mining has also been used in community mining [37].

Generally speaking, both social network analysis and community mining can be seen as graph mining. The community mining can be thought of as sub-graph identification. Previous work on graph mining can be found in [10][34]. Almost all the previous techniques on graph mining and community mining are based on a homogenous graph, *i.e.*, there is only one kind of relationship between the objects. However, in real social networks, there are always various kinds of relationships between the objects. To deal with this problem, we focus in this paper on multi-relational community mining.

Relational mining

Relational mining especially multi-relational data mining has received a lot of attentions in recent years [15][36][29]. Multi-relational data mining aims at dealing with knowledge discovery from relational databases consisting of multiple tables. It tries to analyze data from a multi-relational database directly, without the need to transfer the data into a single table first. Thus the relations mined can reside in a relational or deductive database.

Intuitively, the different tables in relational database can be thought of as different relation networks. Thus, the relation extraction techniques described in this paper has potential applications to multi-relational data mining.

Feature extraction

Feature extraction has received much attention in machine learning and data mining for its usefulness at classification and clustering. Feature extraction can be viewed as finding a linear combination of the original features that can better describe the intrinsic structure of the data set. Typical feature extraction methods include Principle Component Analysis [12], Linear Discriminant Analysis [12], and Locality Preserving Projection [19]. PCA is unsupervised, LDA is supervised, and LPP can be performed under either supervised or unsupervised mode.

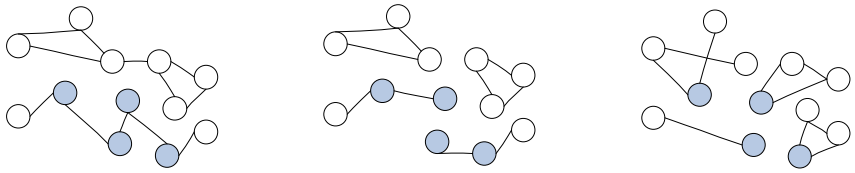
Relation extraction, the problem raised in this paper, is fundamentally related to feature extraction. Given various relations of objects, we aim to find a linear combination of relations which can best reveal the intrinsic relationship between the objects with respect to the user's query. Unfortunately, the state-of-the-art feature extraction methods can hardly be applied to the relation extraction problem, since in relation extraction scenario the explicit vector representations of the objects are not available.

3 Relation Extraction

In this section, we begin with a detailed analysis of the relation extraction problem followed by two algorithms for two cases.

3.1 The Problem

A typical social network likely contains multiple relations. Different relations can be modeled by different graphs. These different graphs reflect the relationship of the objects from different views. For the problems of community mining, these different relation graphs can provide us with different communities.



(a) (b) (c)

Figure 1: There are three relations in the network. The four colored objects are required to belong to the same community, according to a user query.

As an example, the network in Figure 1 may form three different relations. Suppose a user requires the four colored objects belong to the same community. Then we have:

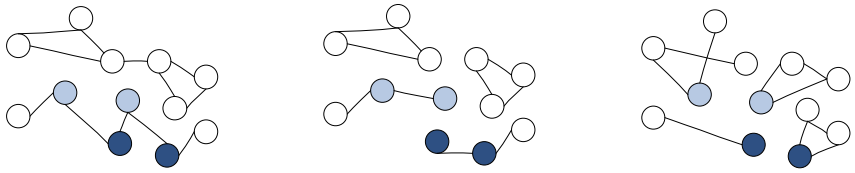
1. Clearly, these three relations have different importance in reflecting the user's information need.

As can be seen, the relation (a) is the most important one, and the relation (b) the second. The relation (c) can be seen as noise in reflecting the user's information need.

2. In the traditional social network analysis, people do not distinguish these relations. The different relations are equally treated. So, they are simply combined together for describing the structure between objects. Unfortunately, in this example, the relation (c) has a negative effect for this purpose. However, if we combine these relations according to their importance, the relation (c) can be easily excluded, and the relation (a) and (b) will be used to discover the community structure, which is consistent with the user's requirement.

3. In the above analysis, the relationship between two objects is considered as a boolean one. The problem becomes much harder if each edge is assigned with a real value weight which indicates to what degree the two objects are related to each other. In such situation, an optimal combination of these relations according to the user's information need cannot be easily obtained.

Different from Figure 1, a user might submit a more complex query in some situations. Take Figure 2 as another example. The relations in the network are the same as those in Figure 1. However, the user example (prior knowledge) changes. The two objects with lighter color and the two with darker color should belong to different communities. In this situation, the importance of these three relations changes. The relation (b) becomes the most important, and the relation (a) becomes the useless (and even negative) one.



(a) (b) (c)

Figure 2: Among the three relations in the network, the two objects with lighter color and the two with darker color should belong to different communities, as user required.

As we can see, in multi-relational social network, community mining should be dependent on the user's example (or information need). A user's query can be very flexible. Since previous community mining techniques only focus on *single* relational network and are independent of the user's query, they cannot cope with such a complex situation.

In this paper, we focus on the relation extraction problem in multi-relational social network. The community mining based on the extracted relation graph is more likely to meet the user's information need. For relation extraction, it can be either linear or nonlinear. Due to the consideration that in real world applications it is almost impossible for a user to provide sufficient information, nonlinear techniques tend to be unstable and may cause over-fitting problems. Therefore, here we only focus on linear techniques.

This problem of relation extraction can be mathematically defined as follows. Given a set of objects and a set of relations which can be represented by a set of graphs $G_i(V, E_i)$, $i = 1, \ldots, n$, where n is the number of relations, V is the set of nodes (objects), and E_i is the set of edges with respect to the *i-th* relation. The weights on the edges can be naturally defined according to the relation strength of two objects. We use M_i to denote the weight matrix associated with G_i, $i = 1, \ldots, n$. Suppose there exists a hidden relation represented by a graph $\widehat{G}(V, \widehat{E})$, and $\widehat{M}$ denotes the weight matrix associated with $\widehat{G}$. Given a set of labeled objects $X = [\mathbf{x}_1, \cdots, \mathbf{x}_m]$ and $\mathbf{y} = [y_1, \cdots, y_m]$ where y_j is the label of $\mathbf{x}_j$ (Such labeled objects indicate partial information of the hidden relation $\widehat{G}$), find a linear combination of the weight matrices which can give the best estimation of the hidden matrix $\widehat{M}$.

3.2 A Regression-Based Algorithm

The basic idea of our algorithm is trying to find an combined relation which makes the relationship between the intra-community examples as tight as possible and at the same time the relationship between the inter-community examples as loose as possible.

For each relation, we can normalize it to make the biggest strength (weight on the edge) be 1. Thus we construct the target relation between the labeled objects as follows:

$$\widetilde{M}_{ij} = \begin{cases} 1, & \text{example } i \text{ and example } j \text{ have} \\ & \text{the same label;} \\ 0, & \text{otherwise.} \end{cases}$$

where $\widetilde{M}$ is a $m \times m$ matrix and $\widetilde{M}_{ij}$ indicates the relationship between examples i and j. Once the target relation matrix is built, we aim at finding a linear combination of the existing relations to optimally approximate the target relation in the sense of L_2 norm. Sometimes, a user is uncertain if two objects belong to the same community and can only provide the possibility that two objects belong to the same community. In such case, we can define $\widetilde{M}$ as follows.

$$\widetilde{M}_{ij} = \text{Prob}(\mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the same community})$$

Let $\mathbf{a} = [a_1, a_2, \cdots, a_n]^T \in R^n$ denote the combination coefficients for different relations. The approximation problem can be characterized by solving the following optimization problem:

$$\mathbf{a}^{opt} = \arg\min_{\mathbf{a}} \|\widetilde{M} - \sum_{i=1}^{n} a_i M_i\|^2 \tag{1}$$

This can be written as a vector form. Since the matrix $M_{m \times m}$ is symmetric, we can use a $m(m-1)/2$ dimensional vector $\mathbf{v}$ to represent it. The problem (1) is equivalent to:

$$\mathbf{a}^{opt} = \arg\min_{\mathbf{a}} \|\tilde{\mathbf{v}} - \sum_{i=1}^{n} a_i \mathbf{v}_i\|^2 \tag{2}$$

Equation (2) is actually a linear regression problem [17]. From this point of view, the relation extraction problem is interpreted as a prediction problem. Once the combination coefficients are computed, the hidden relation strength between any object pair can be predicted.

In real applications, the user does not need to specify the relationships between any pair of objects. That is, the vector $\mathbf{v}$ need not to be $m(m-1)/2$ dimensional. We assume that $\mathbf{v}$ is a k-dimensional vector in the following.

Let us first consider the simplest case that

$$\sum_{i=1}^{n} a_i \mathbf{v}_i = \tilde{\mathbf{v}} \tag{3}$$

We define:

$$V = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n]$$

Thus, equation (3) can be rewritten as follows:

$$V\mathbf{a} = \tilde{\mathbf{v}} \tag{4}$$

Suppose the rank of V is $\min(k, n)$. We have the following facts:

- when $k < n$, the set of solutions to equation (4) forms a $(n-k)$ dimensional vector subspace;
- when $k = n$, there is a unique solution to equation (4);
- when $k > n$, there is no solution to equation (4).

In the first two cases, we get a solution with perfect match (The minimization error is zero). Note that, the value of k reflects the quantity of the user's information needs. k is small when the query submitted by the user is simple.

With a complex query, k can be larger than n. In this case, the optimal solution to (2) is obtained when the derivative of this objective function with respect to $\mathbf{a}$ is zero, i.e.

$$\frac{\partial \|\tilde{\mathbf{v}} - \sum_i^n a_i \mathbf{v}_i\|^2}{\partial a_i} = 0 \quad for \quad i = 1, \cdots, n$$

By some algebraic steps, we have:

$$
\begin{aligned}
& \frac{\partial \|\tilde{\mathbf{v}} - \sum_i^n a_i \mathbf{v}_i\|^2}{\partial a_i} = 0 \\
\Rightarrow \quad & \frac{\partial [(\tilde{\mathbf{v}} - \sum_i^n a_i \mathbf{v}_i)^T (\tilde{\mathbf{v}} - \sum_i^n a_i \mathbf{v}_i)]}{\partial a_i} = 0 \\
\Rightarrow \quad & \frac{\partial (\tilde{\mathbf{v}} - \sum_i^n a_i \mathbf{v}_i)^T}{\partial a_i} (\tilde{\mathbf{v}} - \sum_i^n a_i \mathbf{v}_i) + (\tilde{\mathbf{v}} - \sum_i^n a_i \mathbf{v}_i)^T \frac{\partial (\tilde{\mathbf{v}} - \sum_i^n a_i \mathbf{v}_i)}{\partial a_i} = 0 \\
\Rightarrow \quad & 2 \Big\{ \frac{\partial (\tilde{\mathbf{v}} - \sum_i^n a_i \mathbf{v}_i)}{\partial a_i} \Big\}^T (\tilde{\mathbf{v}} - \sum_i^n a_i \mathbf{v}_i) = 0 \\
\Rightarrow \quad & \mathbf{v}_i^T (\tilde{\mathbf{v}} - \sum_i^n a_i \mathbf{v}_i) = 0 \;\; for \;\; i = 1, \cdots, n \\
\Rightarrow \quad & V^T (\tilde{\mathbf{v}} - \sum_i^n a_i \mathbf{v}_i) = 0 \\
\Rightarrow \quad & V^T (\tilde{\mathbf{v}} - V\mathbf{a}) = 0 \\
\Rightarrow \quad & V^T V \mathbf{a} = V^T \tilde{\mathbf{v}}
\end{aligned}
$$

Since the matrix V has full rank as we assumed, *i.e.*, $\text{rank}(V) = \min(k, n)$, the matrix $V^T V$ is invertible and the optimal solution to (2) is $\mathbf{a}^{opt} = (V^T V)^{-1} V^T \tilde{\mathbf{v}}$.

When the matrix V is rank deficiency, *i.e.*, $\text{rank}(V) < \min(k, n)$, there will be multiple solutions with the same minimization value. In such case, we can choose the $\mathbf{a}$ with minimum norm as our solution [4].

The objective function (2) models the relation extraction problem as an unconstrained linear regression problem. One of the advantages of the unconstrained linear regression is that, it has a close form solution and is easy to compute. However, researches on linear regression problem show that in many cases, such unconstrained least squares solution might not be a satisfactory solution and the coefficient shrinkage technique should be applied based on the following two reasons [17].

1. *Prediction accuracy*: The least-squares estimates often have low bias but large variance [17]. The overall relationship prediction accuracy can sometimes be improved by shrinking or setting some coefficients to zero. By doing so we sacrifice a little bit of bias to reduce the variance of the predicted relation strength, and hence may improve the overall relationship prediction accuracy.

2. *Interpretation*: With a large number of explicit (base) relation matrices and corresponding coefficients, we often would like to determine a smaller subset that exhibit the strongest effects. In order to get the "big picture", we are willing to sacrifice some of the small details.

Such consideration can be shown in the following example. Suppose we have a user query $(o_1, o_2, o_3, o_4, o_5)$, where o_1, o_2, and o_3 belong to one community, but o_4 and o_5 belong to another.

The target relation network can be constructed as:

	o_1	o_2	o_3	o_4	o_5
o_1	$*$	1	1	0	0
o_2	1	$*$	1	0	0
o_3	1	1	$*$	0	0
o_4	0	0	0	$*$	1
o_5	0	0	0	1	$*$

The $*$ in the relation matrix means that we do not consider the self-relation strength. The four basic relation matrices (corresponding to these objects) in the social networks are shown in Figure 3.

	o_1	o_2	o_3	o_4	o_5
o_1	$*$	0.8	0.7	0	0
o_2	0.8	$*$	0.9	0	0
o_3	0.7	0.9	$*$	0	0
o_4	0	0	0	$*$	0.6
o_5	0	0	0	0.6	$*$

(a) Relation M_1

	o_1	o_2	o_3	o_4	o_5
o_1	$*$	0	0.1	0	0
o_2	0	$*$	0	0	0
o_3	0.1	0	$*$	0	0
o_4	0	0	0	$*$	0
o_5	0	0	0	0	$*$

(b) Relation M_2

	o_1	o_2	o_3	o_4	o_5
o_1	$*$	0.1	0	0	0
o_2	0.1	$*$	0	0	0
o_3	0	0	$*$	0	0
o_4	0	0	0	$*$	0.1
o_5	0	0	0	0.1	$*$

(c) Relation M_3

	o_1	o_2	o_3	o_4	o_5
o_1	$*$	0	0	0	0
o_2	0	$*$	0.1	0	0
o_3	0	0.1	$*$	0	0
o_4	0	0	0	$*$	0
o_5	0	0	0	0	$*$

(d) Relation M_4

Figure 3: The four basic relation matrices corresponding to the examples

We can find that $0M_1 + 10M_2 + 10M_3 + 10M_4$ can exactly match the example relation matrix. However, such extracted relation might not a good approximation to the hidden relation on the whole object set. The relation M_1 is more likely to be what the user desires. This is exactly the problem of unconstrained linear regression. We need to use some coefficient shrinkage techniques to solve such problem [17].

Thus, for each relation network, we normalize all the weights on the edges in the range [0, 1]. And, we put a constraint $\sum_{i=1}^{n} a_i^2 \leq 1$ on the objective function (2). Finally, our algorithm tries

to solve the following minimization problem,

$$\begin{aligned} \mathbf{a}^{opt} &= \arg\min_{\mathbf{a}} \|\tilde{\mathbf{v}} - \sum_{i=1}^{n} a_i \mathbf{e}_i\|^2 \\ &\text{subject to } \sum_{i=1}^{n} a_i^2 \leq 1 \end{aligned} \tag{5}$$

Such a constrained regression is called *Ridge Regression* [17] and can be solved by some numerical methods [4]. When we use such constrained relation extraction, the coefficients of the extracted relation for the above example are 1, 0, 0, 0. This shows that our constrained relation extraction can really solve the problem.

3.3 A MinCut-Based Algorithm

In the last subsection, we have presented a general method for exacting the hidden relation based on regression model. However, this method may fail when the examples provided by the user belong to only one community, which is referred to *single community issue* in the rest of this paper. We provide an intuitive example in the following.

Suppose we have a user query $(o_1, o_2, o_3, o_4, o_5)$, which belong to the same community. In the following two relations shown in Fig. 4(a) and Fig. 4(b), regression model would prefer the relation M_1, since the higher connectivity between o_1, o_2, o_3, o_4 achieves a lower square error to the target relation. However, in relation M_1, the connectivity between o_5 and the other four examples are very weak. As can be seen, the connectivity in M_2 is much more uniform than that in M_1 while it has comparable strength. Therefore, M_2 should be a better choice for this user query. Unfortunately, the square error of M_2 is larger than that of M_1. This shows that the regression model may fail in such a case.

	o_1	o_2	o_3	o_4	o_5
o_1	$*$	0.9	0.8	1	0.1
o_2	0.9	$*$	1	0.9	0.1
o_3	0.8	1	$*$	1	0.1
o_4	1	0.9	1	$*$	0.1
o_5	0.1	0.1	0.1	0.1	$*$

(a) Relation M_1

	o_1	o_2	o_3	o_4	o_5
o_1	$*$	0.4	0.4	0.5	0.5
o_2	0.4	$*$	0.3	0.2	0.4
o_3	0.4	0.3	$*$	0.3	0.3
o_4	0.5	0.2	0.3	$*$	0.4
o_5	0.5	0.4	0.3	0.4	$*$

(b) Relation M_2

Figure 4: Two existing relations

In order to deal with the *single community issue*, we need to take into account the weakest connection in the extracted relation. By graph theory, the value of the *minimum cut* on the graph

can be used to evaluate the tightness of the graph.

Let G denote a weighted graph with weight matrix M. Let m denote the number of vertices. A cut on the graph is defined as a set of edges which separates the vertices into two disconnected groups denoted by A and B such that $A \cap B = \phi$ and $A \cup B = G$. Thus, the value of the cut is:

$$cut(G) = \sum_{i \in A} \sum_{j \in B} M(i,j)$$

It is easy to see that there are totally $2^m - 2$ different cuts. Let $cut_k(G) = (A_k, B_k)$ denote the k-th cut. The minimum cut is defined as:

$$mincut(G) = \min_k \{cut_k(G)\}$$

If a graph can be easily cut into two subgraphs, it has a small minimum cut value. As an extreme case, the minimum cut value of a disconnected graph is 0. Naturally, the optimal extracted relation graph should have a large minimum cut value. Thus, for single community issue, we try to extract the optimal relation graph by maximizing its minimum cut value.

Let G_i, $i = 1, \ldots, n$, denote the existing relation graphs defined only on the user query examples and M_i denote the corresponding weight matrices. Let $\mathbf{a} = [a_1, a_2, \cdots, a_n]^T \in R^n$ denote the combination coefficients for different graphs. Thus $M = \sum_{i=1}^n a_i M_i$ is the weight matrix of the combined relation graph G. Let $mincut(G)$ denote the minimum cut value of G. Our objective function can be written as follows:

$$\mathbf{a}^{opt} = \arg\max_{\mathbf{a}} \{mincut(\sum_{i=1}^{n} a_i G_i)\} \tag{6}$$

Generally, the minimum cut problem is an NP-hard problem. Thus the optimization problem (6) cannot be easily solved. However, in our problems, the number of examples provided by the user is usually small. That is, m is small, typically less than 10. Thus we can use linear programming techniques to solve the optimization problem (6) by the following derivation:

$$\begin{aligned}
mincut(G) &= \min_{1 \le k \le 2^m - 2} \{cut_k(G)\} \\
&= \min_{1 \le k \le 2^m - 2} \{ \sum_{i \in A(k)} \sum_{j \in B(k)} M(i,j) \} \\
&= \min_{1 \le k \le 2^m - 2} \{ \sum_{i \in A(k)} \sum_{j \in B(k)} (\sum_{l=1}^{n} a_l M_l(i,j)) \} \\
&= \min_{1 \le k \le 2^m - 2} \{ \sum_{l=1}^{n} a_l (\sum_{i \in A(k)} \sum_{j \in B(k)} M_l(i,j)) \} \\
&= \min_{1 \le k \le 2^m - 2} \{ \sum_{l=1}^{n} a_l \cdot cut_k(G_l) \}
\end{aligned}$$

Let $v = mincut(G)$. The optimization problem in Eq. (6) can be reduced to the following linear programming problem:

$$
\begin{aligned}
& \max \quad v \\
st. \quad & \sum_{l=1}^{n} a_l \cdot cut_k(G_l) - v \geq 0, \quad (1 \leq k \leq 2^m - 2) \quad (*) \\
& \sum_{l=1}^{n} a_l = 1 \\
& a_l \geq 0, \quad (1 \leq l \leq n)
\end{aligned}
$$

With the constraints (*), v is guaranteed to be the minimum cut value, and by maximizing v we can obtain the optimal combination coefficients a_i. The number of constraints in this problem is $2^m - 2 + n + 1$, where m is the number of user-provided examples which is usually less than 10, and n is the number of existing relations. The above problem can be efficiently solved by linear programming techniques [3].

The proposed regression based algorithm and the MinCut based algorithm are used under different situations. When a user provides multiple community examples, regression-based algorithm can be used to find the best combination; when he provides single community examples, MinCut-based algorithm can be used.

4 Community Mining on the Extracted Relation

There are many community mining algorithms proposed for different scenarios [16][13][9][37]. However, all these algorithms focus on a homogenous graph. Based on the relation extraction algorithm we proposed, we can extract the hidden relation based on the examples (queries) provided by the user. Thus, the heterogeneous social network (multiple explicit relations) becomes a homogeneous one. Any previous community mining algorithm can be then applied and expected to produce good results.

In this section, we briefly describe two methods for community mining. These two methods will be used in our experiments.

4.1 Normalized Cut

In some cases, the number of communities in the social network is available, and each object belongs to a unique community. In some sense, the community mining problem is very similar to the clustering problem. Thus, graph-based clustering algorithm can be used. We choose the *Normalized Cut* [32][27] due to its excellent performance and solid theoretical foundation. Suppose there are n objects and k communities, and the relation network was represented as a symmetric matrix $S \in R^{n \times n}$. The algorithm can be stated as follows:

1. Define D to be a diagonal matrix whose (i, i)-element is the sum of S's i-th row, and construct the matrix $L = D^{-1/2}SD^{-1/2}$.

2. Find $y_1, y_2, \ldots, y_k$, the k largest eigenvector of L, and form the matrix $Y = [y_1 y_2 \cdots y_k] \in R^{n \times k}$ by stacking the eigenvectors in columns.

3. Treat each row of Y as a point in R^k, and cluster them into k clusters via K-means.

4. For each cluster, we define it as a community.

It would be important to note that Normalized Cut is an extension of the standard minimum cut algorithm. Please see [32] for more details.

4.2 Threshold Cut

In some cases, the user expects the mined community has a reasonable size, and relation strength in the mined community is strong enough. A simple method based on predefined threshold can be used for this purpose. We name this method as *Threshold Cut*. There are two ways to perform this algorithm.

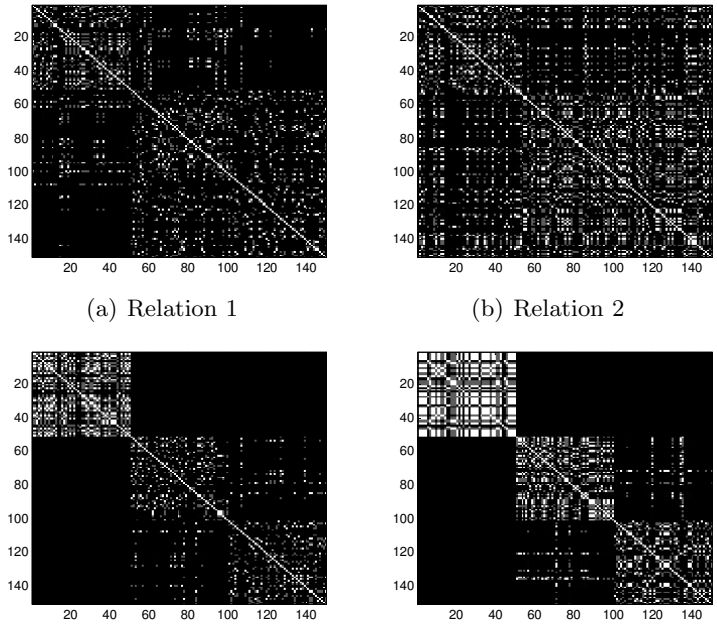
1. User provides a threshold and expects that the relation strength between any pair of objects in any community is no less than this threshold. To do so, we remove the edges whose weight is less than the threshold. Finally, all the connected subgraphs (with more than one object) are taken as communities.

2. User expects a certain number of objects appear in all the mined communities. In this situation, we initialize the threshold as the largest edge weight in the graph. The number of objects that belong to some communities will increase as the threshold reduces. When the number of objects satisfies the user's need, the corresponding threshold is set, and the communities are mined out.

This method is especially suitable for finding communities with large relation strength in the network.

5 Experiments

In this section, we present our experimental study of the proposed relation extraction algorithm on the Iris and DBLP datasets. On the Iris dataset, we give quantitative results. On the DBLP dataset, some interesting examples are provided to show the effectiveness of our proposed algorithm.



(a) Relation 1 (b) Relation 2

(c) Relation 3 (d) Relation 4

Figure 5: The four synthetic relation networks on Iris dataset

5.1 Synthetic Relation Networks on Iris Data

In this section, we use the Iris data set to verify our algorithm. The iris dataset, popularly used for testing clustering and classification algorithms, is taken from UCI ML repository[1]. It contains 3 classes of 50 instances each, where each class refers to a type of Iris plant. Each instance has four features, out of which it is known that F_3 (petal length) and F_4 (petal width) are more important for the underlying clusters.

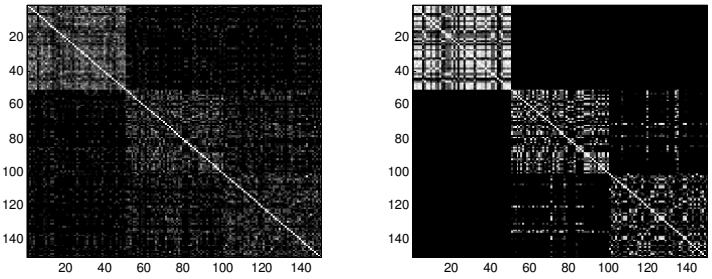
For each feature F_r, we constructed a relation network $M_{r,ij}$ as follows:

$$M_{r,ij} = e^{-(x_i - x_j)^2} \tag{7}$$

Thus, the iris data can be viewed as a multi-relational social network with three hidden communities. The four relation matrices M_1, M_2, M_3, and M_4, constructed from the four features independently, were shown in Figure 5. The brightness reflects the relation strength between two objects.

The following experiment was designed as given some labeled data as user query, using the regression based relation extraction algorithm described in Section 3.2 to extract the relation. Community mining was then applied on the extracted relation.

[1]http://www.ics.uci.edu/∼mlearn/MLRepository.html



(a) Baseline Relation (b) Extracted Relation

Figure 6: The baseline relation and extracted relation

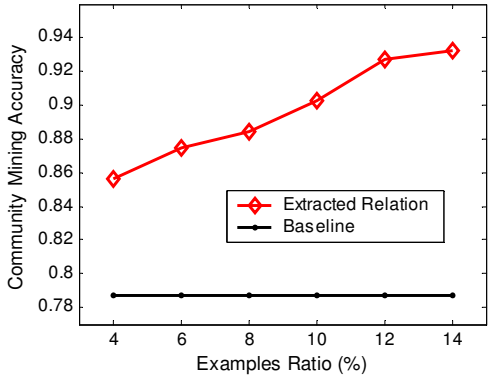


Figure 7: Community mining accuracy on the extracted relation

5.1.1 Evaluation using community mining

In this experiment, we applied the Normalized Cut [32] algorithm described in previous section as our community mining algorithm.

The performance of community mining result is evaluated by comparing the obtained label of each object with the ground truth. Given an object $\mathbf{x}_i$, let r_i and s_i be the obtained community label and the ground truth, respectively. The accuracy AC is defined as follows:

$$AC = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n} \tag{8}$$

where n is the total number of objects and $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and map(r_i) is the permutation mapping function that maps each community label r_i to the equivalent label from the ground truth. The best mapping can be found by using the Kuhn-Munkres algorithm [25].

5.1.2 Results

As described in Section 3, the extracted relation can be defined as,

$$M' = \sum_{i=1}^{4} a_i M_i \tag{9}$$

Traditional community mining algorithms are independent of the user-submitted query. Thus, the four relations are treated equally, *i.e.*, $a_i = 0.25$, $i = 1, \ldots, 4$. Community mining is then performed on this combined graph $M' = \sum_{i=1}^{4} 0.25 M_i$. We take this as the baseline. With the user's query, the relation extraction algorithm described in section 3 can be applied to extract a combined relation which respects the user's query. One can expect better performance with more labeled examples.

For a given number k, we randomly selected k examples for each class as the user query and extracted the optimal relation. The Normalized Cut algorithm was applied on the extracted relation to mine the communities and the accuracy was recorded. This process was repeated 100 times and the average performance was computed.

Figure 6 shows the baseline relation and the extracted relation. The extracted relation is obtained with 10% examples provided and the combination coefficients for the four base relations are 0, 0, 0.5948 and 0.8039, respectively. The community mining accuracy on extracted relation based on different examples ratio is shown in Figure 7.

This experiment shows that our algorithm can extract the optimal relation based on a few examples (labeled data). When some label information is available, it is always the case that a better relation can be extracted. In a multi-relational network, the user's information need can be extremely diverse. This particularly makes relation extraction an important pre-processing for social network analysis.

5.2 Mining Hidden Networks on the DBLP Data

In this part, we present our experimental results based on DBLP (Digital Bibliography & Library Project) data. The DBLP server (http://dblp.uni-trier.de/) provides bibliographic information on major computer science journals and proceedings. It indexes more than 500000 articles and more than 1000 different conferences (by May 2004).

Taking the authors in DBLP as objects, there naturally exist multiple relations between them. Authors publish paper in difference conferences. If we treat that authors publish paper(s) in the same conference as one kind of relation, these 1000 conferences provide us 1000 different relations. Given some examples (*e.g.*, a group of authors), our experiment is to study how to extract a new relation using such examples and find all the other groups in the relation. The extracted relation can be interpreted as the groups of authors that share a certain kind of similar interests.

5.2.1 Data preparation and graph generation

The DBLP server provides all the data in the XML format as well a simple DTD. We extracted the information of author, paper and conference. There are several points we should mention:

1. In the DBLP data, each author (researcher) is represented by a string. We do not distinguish two different authors carrying the same name. We believe that the duplicate names are only a tiny part of the whole dataset, thus will not be a problem in our experiment.

2. DBLP data does not provide the proceeding information for all the papers. We simply use the "key" attribute (usually in the format "conf/kdd/TsumotoT95a") to extract the conference identifier "KDD" and append with the "year" attribute. We assume that all papers in the same proceeding should have the same path name in the "key" attribute and the "year" attribute. This may not be true in all cases. However, we believe this is good enough for our experiments.

We generate different kinds of graphs (social networks) based on the extracted information. For each proceeding, we construct a graph with researchers as the nodes, which is called *proceeding graph* thereafter. If two researchers have paper(s) in this proceeding, the edge between the two corresponding nodes is set to 1. Otherwise, it is set to 0. For each conference, we add up the proceeding graphs of the same conference over years, which is called *conference graph* thereafter. Finally, we choose the top 70 conference graphs based on the number of distinct authors in that conference.

Every conference graph reflects the relationship between the researchers pertaining to a certain research area. Generally, if two researchers are connected by an edge in the conference graph, they may share the same research interests.

For each graph, we normalize the edge weight by dividing the maximum weight in the whole graph. The resulting weight has a range [0, 1]. The greater the weight is, the stronger the relation is.

5.2.2 Experiment results

In this experiment, we provide the system with some queries (some groups of researchers) to examine if our algorithm can capture the hidden relation between the researchers. The query examples are believed to belong to the same community, thus the MinCut based relation extraction algorithm was used. When the hidden relation was extracted, the community finding algorithm described in Section 4 was applied to find the communities on the new network.

Experiment 1. In the first case, there are two queries provided by the user.

1. Philip S. Yu, Rakesh Agrawal, Hans-Peter Kriegel, Padhraic Smyth, Bing Liu, Pedro Domingos.

2. Philip S. Yu, Rakesh Agrawal, Hans-Peter Kriegel, Hector Garcia-Molina, David J. DeWitt, Michael Stonebraker.

Both of the two queries contain 6 researchers. The first three researchers are the same in the two queries.

Table 1: Coefficients of different conference graphs for two queries (sorted on the coefficients)

Query 1		Query 2	
Conference	Coefficient	Conference	Coefficient
KDD	1	SIGMOD	0.528
		ICDE	0.262
		VLDB	0.210

Table 1 shows the coefficients of the extracted relation for the two queries. KDD is a data mining conference, and high weight on the KDD graph indicates the common interest on data mining. On the other hand, SIGMOD, VLDB and ICDE are three database conferences. High weights on these conference graphs indicate the common interest on database area. The extracted relation for query 1 has KDD graph with weighting 1, which tells us that the researchers in query 1 share common interest on data mining. For query 2, the extracted relation tells us those researchers share common interest on database.

Table 2: Researchers' activities in conferences

Researcher	KDD	ICDE	SIGMOD	VLDB
Philip S. Yu	7	15	10	11
Rakesh Agrawal	6	10	13	15
Hans-Peter Kriegel	7	9	11	8
Padhraic Smyth	10	1	0	0
Bing Liu	8	1	0	0
Pedro Domingos	8	0	2	0
Hector Garcia-Molina	0	15	12	12
David J. DeWitt	1	4	20	16
Michael Stonebraker	0	12	19	15

While we examine the publication of these researchers on these four conferences as listed in Table 2, we clearly see the extracted relation really captures the semantic relation between the researchers in the queries.

Furthermore, with the extracted relation graph, we applied the community mining algorithm *threshold cut* and obtained the corresponding communities. For each query, we list one example community below:

1. Community for query 1: Alexander Tuzhilin, Bing Liu, Charu C. Aggarwal, Dennis Shasha, Eamonn J. Keogh, ...

2. Community for query 2: Alfons Kemper, Amr El Abbadi, Beng Chin Ooi, Bernhard Seeger, Christos Faloutsos, ...

Let us see what will happen if we only submit the first three names in one query. The extracted relation is shown in Table 3. The extracted relation really captures the two areas (data mining and dababase) in which these researchers are interested.

Table 3: Combined Coefficients

Conference Name	Coefficient
SIGMOD	0.302
KDD	0.279
ICDE	0.217
VLDB	0.202

Experiment 2. Let us try another example. The two queries are:

1. Pat Langley, Andrew W. Moore, Michael J. Pazzani, James P. Callan, Yiming Yang, Thomas G. Dietterich

2. Pat Langley, Andrew W. Moore, Michael J. Pazzani, Raymond T. Ng, Philip S. Yu

The extracted relations are shown in Table 4. And the activities of these researchers on these conference are shown in Table 5. In this case, we can draw the same conclusion as the first case. The relation extraction algorithm proposed in this paper can really extract the hidden relation from the user-provided examples.

Table 4: Coefficients of different conference graphs for two queries (sorted on the coefficient)

Query 1		Query 2	
Conference	Coefficient	Conference	Coefficient
ICML	1	KDD	0.671
		ICML	0.329

From the above two experiments, one can see that data mining (KDD) is really an interdisciplinary area. Many researchers active in data mining are also active in database and machine learning. Our relation extraction algorithm captures the hidden relation among the researchers provide by users.

Table 5: Researchers' activities in conferences

Researcher	ICML	KDD	SIGIR	SIGMOD
Pat Langley	11	4	0	0
Andrew W. Moore	10	3	0	0
Michael J. Pazzani	10	6	1	1
James P. Callan	3	0	10	1
Yiming Yang	5	1	8	0
Thomas G. Dietterich	12	1	0	0
Raymond T. Ng	0	8	0	7
Philip S. Yu	1	7	0	10

Experiment 3. In this case, four researchers mainly focus on different areas were submitted as the query. They are Avideh Zakhor, Lars Erik Holmquist, Elisa Bertino, and Makoto Sato. Based on the statistical information in the DBLP data, Avideh Zakhor focuses on ICIP, ISCAS, DCC, and CVPR; Lars Erik Holmquist on HUC, CHI, IWEC, and HCI; Elisa Bertino mostly on RIDE, ICDE, DBSEC, TIME, EDBT, and SIGMOD; whereas Makoto Sato mainly on VR.

Our algorithm extracted the hidden relation between them from the DBLP data. To our surprise, SIGGRAPH was selected as the hidden relation. When we carefully examined the DBLP statistics, we found that all these four researchers really showed up once in SIGGRAPH.

6 Potential Applications

With the rapid growth of Internet, heterogeneous social networks become ubiquitous. Here we discuss some potential applications of relation extraction. Based on our view, the heterogeneous social networks on the web can be categorized as follows.

Social Community Websites. Social community websites are not so prosperous until very recently. Examples of social community websites include www.orkut.com and xanga.com. As of Nov.1, 2004, www.orkut.com has attracted more than 2.2 million registered users, and it is still growing very fast. On such websites, people always register by referral from friends. Thus all the registered users form a big social graph. Users can create different friend links based on their familiarity with their friends. They can form all kinds of communities. Friends usually share some kind of community. For example, a person may have some friends that are his high school classmates, some his university classmates, some his colleagues in a company, and some sharing the same interest like hiking or swimming. Classmates, colleagues, and hobbyists form communities. Thus such a website naturally supports heterogeneous social networks.

Forums and Newsgroups. Forums and newsgroups are popular places where people discuss all kinds of topics with other people. They also represent multiple social networks. For example, on

the famous forum www.craigslist.org, people participate in discussions on housing, jobs, interests, neighborhood, second-handed goods, and other kinds of topics. The active discussions among a group of people on a topic represent some kind of relationship among them.

Instant Messengers and Emails. Instant Messengers (IMs) and Emails are two means that people contact others. Instant Messengers might be one of the most popular tool today that a lot of people use every day. MSN Messenger, Yahoo! Messenger, AOL Instant Messenger, ICQ are among the most popular ones. People usually have a friend list on the IM and they always manage their friends in groups. The default group set up of MSN Messenger includes Buddies, Coworkers, Family, Friend and Others. These group provides valuable information on the semantics of the relationship. Such information can be utilized in the multi-relational network analysis. Emails are another kind of data can also be used to mine the relationship among people. Some previous works already started on such direction [31].

These multi-relational social networks provide us with great opportunities to apply the relation extraction algorithm for application exploration, as briefly listed below.

1. **Friend suggestion.** A lot of people join those web-based social communities in hope of finding a boy/girl friend, a soul mate, some friends for picnic, a business partner, and so on. Given the large degree of each node in a social network and the exponential increase of candidates with the length of the friend link, it often takes much time to find the people they want. In fact, the *people they want* is a high-level semantic which can be considered as a combination of multiple underlying social networks. Our model provides a possible solution: The system collects examples from the user's input and trains the weights for each underlying network. The trained combination can then be used to do friend suggestion.

2. **Targeted marketing.** Targeted marketing, such as viral marketing and content-based advertisement, are popular both in research and in business. By mining multi-relational social networks, we are able to give more precise suggestion on targeted marketing. For example, people buying different kinds of products or participating different kinds of activities may form multiple social networks. When a new product is rolling out, it is easy to find potential customers based on the comprehensive evaluation of such multiple social networks.

3. **Network prediction.** Modelling the evolution of social network has been actively studied in recent years. However, all existing approaches focus on single social networks. But in reality, one social network may affect the evolution of another. For example, the marriage relationship might be strongly related to some underlying social networks like classmates, colleagues or people with the same interests. However, the weights of these underlying networks may vary with times and countries. Thus, these weights can be used for predicting marriage. One may also observe how the marriage relationship could be affected by the evolution of several related social networks.

The center idea of all these applications comes from the same problem: *how to obtain the hidden high-level relation network given the underlying low-level social networks*. Our relation extraction solution, though simple, offers the first plausible tool to tackle this problem.

7 Discussion

Since mining hidden communities in heterogeneous networks represents a promising research direction, there are many issues that need to be discussed. Here we focus on the problem solving philosophy.

First, one may *wonder the complexity at comprehension and combination of multiple social networks in the analysis*. We do agree that multiple social networks form complex, multiple, interrelated graphs, and with the massive amount of data mounting, it is challenging for anyone to grasp the whole picture of such dynamic, evolving social networks and work out a balanced combination of multiple networks for a particular user query. However, such multiple networks do exist, and it is inappropriate to blindly merge them into one since different networks plays different roles in particular queries, as shown in our experiments. Therefore, we believe that developing new multi-network mining algorithms to dynamically combine multiple relevant networks to form combined "virtual" networks based on user's example queries is a new and appropriate problem-solving methodology.

Second, since it is difficult for a user to comprehend the whole picture of numerous social networks, one may wonder *how a user is able to pose high-quality queries*. Based on our experience, although it is difficult for a user to comprehend the overall multiple networks, a user usually has good knowledge on a small set of examples (such as influential researchers, movie/sport stars, big companies, or popular commodities). Such firm grasp of a small set of examples is often sufficient to pose intelligent queries, learn additional facts, and form informative combined networks. This has been also demonstrated in our DBLP experiments.

Third, one may wonder *how to comprehend the answers returned from such a network analysis*. Since a derived hidden network is a weighted matrix as a combination of multiple existing networks, it is often difficult to understand the minor weight differences in the results. However, the real essence is at the new facts derived from such hidden networks and their associated rankings. This resembles Google-like keyword-based Web search. It is not so crucial to understand the derived Web linkage weighting and claim it is optimal. However, the return of quality rankings on the interesting results demonstrate its utility.

8 Conclusions

Different from most social network analysis studies, we assume that there exist multiple, heterogeneous social networks, and the sophisticated combinations of such heterogeneous social networks may generate important new relationships that may better fit user's information need. Therefore, our approach to social network analysis and community mining represents a major shift in methodology from the traditional one, a shift from single-network, user-independent analysis to multi-network, user-dependant, and query-based analysis. Our argument for such a shift is clear: multiple, heterogeneous social networks are ubiquitous in the real world and they usually *jointly* affect people's social activities.

Based on such a philosophy, we worked out a new methodology for relation extraction, and proposed two algorithms in different situations. With such query-dependent relation extraction and community mining, fine and subtle semantics are captured effectively. Our experimental results on Iris data set and DBLP data set demonstrate the effectiveness of our algorithm since it substantially improves prediction accuracy in comparison with the baseline approach and it convincingly discovers interesting relations and communities. Our discussion also shows it is expected that the query-based relation extraction and community mining would give rise to a lot of potential new applications in social network analysis.

Our study is the first one that promotes query-based mining of heterogeneous social networks. There are a lot of issues that need to be studied further.

First, our approach adopts a regression-based graph matrix analysis approach. There are potentially many other approaches that can be explored and compared with this approach. We will expect that future studies may propose even more powerful approaches in relation extraction than what is proposed here.

Second, our relation extraction algorithm has made a lot of simplifications in the analysis. In general, links within the same network and among different networks may carry different weights. For example, one can imagine that the links among coauthor networks should be inherently stronger than those among co-proceedings since average size (# of links) in the coauthor group is much smaller than that in the co-proceedings group. This is not considered in our simple model. Thus we expect the prediction power will be substantially enhanced if such information is incorporated in the new algorithm.

Third, our query model considers only one simple group of nodes (such as researchers). A more powerful query model may involve and, or, not operators on those groups. For example, one may like to find those who co-attend the same conference but *never* co-authored a paper using the not operator. This will be useful for finding referees for conference submissions.

These issues may form an exciting frontier for future research.

References

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. Technical report, Xerox Parc, 2002.

[2] R. Agrawal, S. Rajagopalan, R. Srikant, and Y. Xu. Mining newsgroups using networks arising from social behavior. In *Proceedings of 12th International World Wide Web Conference*, 2003.

[3] M. Bazaraa, J. Jarvis, and H. Sherali. *Linear Programming and Network Flows*. Wiley, 3rd edition edition, 2004.

[4] A. Bjorck. *Numerical Methods for Least Squares Problems*. SIAM, 1996.

[5] D. Cai, X. He, J.-R. Wen, and W.-Y. Ma. Block-level link analysis. In *Proceedings of the ACM SIGIR'2004*, 2004.

[6] S. Chakrabarti. Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. In *Proceedings of the 10th International World Wide Web Conference*, 2001.

[7] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. In *Proceedings of The 8th International World Wide Web Conference*, 1999.

[8] C. Chen and L. Carr. Trailblazing the literature of hypertext: Author co-citation analysis (1989-1998). In *Proceedings of the 10th ACM Conference on Hypertext and hypermedia*, 1999.

[9] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *Proceedings of the 17th International Conference on Machine Learning*, 2000.

[10] D. J. Cook and L. B. Holder. Graph-based data mining. *IEEE Intelligent Systems*, 15(2):32–41, 2000.

[11] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66. ACM Press, 2001.

[12] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, Hoboken, NJ, 2nd edition, 2000.

[13] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD-2000)*, 2000.

[14] G. W. Flake, S. Lawrence, C. L. Giles, and F. Coetzee. Self-organization of the web and identification of communities. *IEEE Computer*, 35(3):66–71, 2002.

[15] J. Gehrke, R. Ramakrishnan, and V. G. Rainforest. A framework for fast decision tree construction of large datasets. In *Proc. 1998 Int. Conf. Very Large Data Bases (VLDB'98)*, 1998.

[16] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*, 1998.

[17] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[18] T. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th International World Wide Web Conference*, May 2002.

[19] X. He and P. Niyogi. Locality preserving projections. *Advances in Neural Information Processing Systems*, 16, 2003.

[20] H. Kautz, B. Selman, and A. Milewski. Agent amplified communication. In *Proceedings of AAAI-96*, pages 3–9, 1996.

[21] H. Kautz, B. Selman, and M. Shah. Referral web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.

[22] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–622, 1999.

[23] J. M. Kleinberg. Hubs, authorities, and communities. *ACM Computing Surveys*, 31(4), 1999.

[24] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber communities. In *Proceedings of The 8th International World Wide Web Conference*, 1999.

[25] L. Lovasz and M. Plummer. *Matching Theory*. Akadémiai Kiadó, North Holland, Budapest, 1986.

[26] S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[27] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, 2002.

[28] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[29] Parag and P. Domingos. Multi-relational record linkage. In *Proc. the KDD-2004 Workshop on Multi-Relational Data Mining*, 2004.

[30] A. Popescul, G. W. Flake, S. Lawrence, L. H. Ungar, and C. L. Giles. Clustering and identifying temporal trends in document databases. In *IEEE Advances in Digital Libraries*, 2000.

[31] M. F. Schwartz and D. C. M. Wood. Discovering shared interests using graph analysis. *Communications of the ACM*, 36(8):78–89, 1993.

[32] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on PAMI*, 22(8):888–905, 2000.

[33] M. Toyoda and M. Kitsuregawa. Observing evolution of web communities. In *Proceedings of the 11th International World Wide Web Conference (WWW2002)*, 2002.

[34] T. Washio and H. Motoda. State of the art of graph-based data mining. *SIGKDD Explor. Newsl.*, 5(1):59–68, 2003.

[35] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, UK, 1994.

[36] X. Yin, J. Han, J. Yang, and P. S. Yu. Crossmine: Efficient classification across multiple database relations. In *Proc. 2004 Int. Conf. on Data Engineering (ICDE'04)*, 2004.

[37] W.-J. Zhou, J.-R. Wen, W.-Y. Ma, and H.-J. Zhang. A concentric-circle model for community mining. Technical report, Microsoft Research, 2002.