# Community Mining from Multi-relational Networks*

Deng Cai[1], Zheng Shao[1], Xiaofei He[2], Xifeng Yan[1], and Jiawei Han[1]

[1] Computer Science Department, University of Illinois at Urbana Champaign
{dengcai2, zshao1, xyan, hanj}@cs.uiuc.edu
[2] Computer Science Department, University of Chicago
xiaofei@cs.uchicago.edu

**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. 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.

## 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) [1][2]. 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 [3] and evaluating the importance of individuals [4][5]. In a typical social network, there always exist various relationships between individuals, such as friendships, business relationships, and common interest relationships.

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

* 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.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 445–452, 2005.

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.

Such a problems 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.

The rest of this paper is organized as follows. Section 2 presents our algorithm for relation extraction. The experimental results on the DBLP data set are presented in Section 3. Finally, we provide some concluding remarks and suggestions for future work in Section 4.

## 2 Relation Extraction

In this section, we begin with a detailed analysis of the relation extraction problem followed by the algorithm.

### 2.1 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.

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.

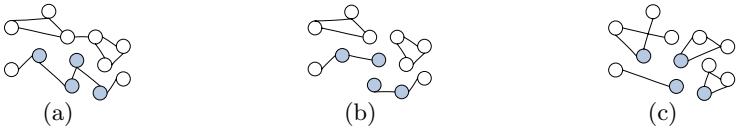


**Fig. 1.** There are three relations in the network. The four colored objects are required to belong to the same community, according to a user query.

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.

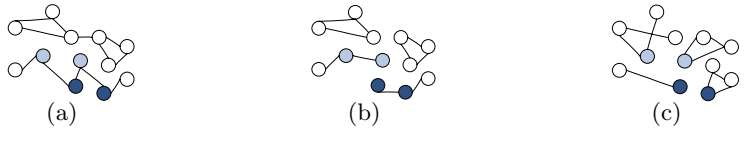


**Fig. 2.** Among the three relations in the network, the two objects with lighter color and the two with darker color should belong to different communities, as user required

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 $\widehat{M}$.

### 2.2 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:

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

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 [6]. 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. There are many efficient algorithms in the literature to solve such a regression problem [7].

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 [6].

1. *Prediction accuracy*: The least-squares estimates often have low bias but large variance [6]. 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.

Our technical report [8] provides an example to explain such consideration. This problem can be solved by using some coefficient shrinkage techniques [6].

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,

$$\begin{aligned} \mathbf{a}^{opt} &= \arg\min_{\mathbf{a}} \|\tilde{\mathbf{v}} - \sum_{i=1}^{n} a_i \mathbf{e}_i\|^2 \\ &\text{subject to } \sum_{i=1}^{n} a_i^2 \leq 1 \end{aligned} \tag{3}$$

Such a constrained regression is called *Ridge Regression* [6] and can be solved by some numerical methods [7]. 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. For more details on our relation extraction algorithm, please refer to [8].

## 3 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.

### 3.1 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.

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.

### 3.2 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. We only show one example in this paper, please refer to [8] for more example queries.

**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 | 0.949 | SIGMOD | 0.690 |
| SIGMOD | 0.192 | ICDE | 0.515 |
| ICDE | 0.189 | VLDB | 0.460 |
| VLDB | 0.148 | KDD | 0.215 |

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 |

**Table 3.** Combined Coefficients

| Conference Name | Coefficient |
|---|---|
| SIGMOD | 0.586 |
| KDD | 0.497 |
| ICDE | 0.488 |
| VLDB | 0.414 |

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* [8] 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.

## 4 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 and a new algorithm for relation extraction. With such query-dependent relation extraction and community mining, fine and subtle semantics are captured effectively. 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.

## References

1. Milgram, S.: The small world problem. Psychology Today **2** (1967) 60–67
2. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge, UK (1994)
3. Schwartz, M.F., Wood, D.C.M.: Discovering shared interests using graph analysis. Communications of the ACM **36** (1993) 78–89
4. Kautz, H., Selman, B., Milewski, A.: Agent amplified communication. In: Proceedings of AAAI-96. (1996) 3–9
5. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press (2001) 57–66
6. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning. Springer-Verlag (2001)
7. Bjorck, A.: Numerical Methods for Least Squares Problems. SIAM (1996)
8. Cai, D., Shao, Z., He, X., Yan, X., Han, J.: Mining hidden community in heterogeneous social networks. Technical report, Computer Science Department, UIUC (UIUCDCS-R-2005-2538, May, 2005)