Generating Social Network Features for Link-Based Classification

Jun Karamon[1], Yutaka Matsuo[2], Hikaru Yamamoto[3], and Mitsuru Ishizuka[1]

[1] The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
karamon@mi.ci.i.u-tokyo.ac.jp,ishizuka@i.u-tokyo.ac.jp
[2] National Institute of Advanced Industrial Science and Technology,
1-18-13 Soto-kanda, Chiyoda-ku, Tokyo, Japan
y.matsuo@aist.go.jp
[3] Seikei University, 3-3-1 Kichijoji Kitamachi, Musashino-shi, Tokyo, Japan
yamamoto@econ.seikei.ac.jp

Abstract. There have been numerous attempts at the aggregation of attributes for relational data mining. Recently, an increasing number of studies have been undertaken to process social network data, partly because of the fact that so much social network data has become available. Among the various tasks in link mining, a popular task is *link-based classification*, by which samples are classified using the relations or links that are present among them. On the other hand, we sometimes employ traditional analytical methods in the field of social network analysis using e.g., centrality measures, structural holes, and network clustering. Through this study, we seek to bridge the gap between the aggregated features from the network data and traditional indices used in social network analysis. The notable feature of our algorithm is the ability to invent several indices that are well studied in sociology. We first define general operators that are applicable to an adjacent network. Then the combinations of the operators generate new features, some of which correspond to traditional indices, and others which are considered to be new. We apply our method for classification to two different datasets, thereby demonstrating the effectiveness of our approach.

1 Introduction

Recently, increasingly numerous studies have been undertaken to process network data (e.g., social network data and web hyperlinks), partly because of the fact that such great amounts of network data have become available. *Link mining* [6] is a new research area created by the intersection of work in link analysis, hypertext and web mining, relational learning, and inductive logic programming and graph mining. A popular task in link mining is *link-based classification*, classifying samples using the relations or links that are present among them. To date, numerous approaches (e.g. [8]) have been proposed for link-based classification, which are often applied to social network data.

A social network is a social structure comprising nodes (called *actors*) and relations (called *ties*). Prominent examples of recently studied social networks

J.N. Kok et al. (Eds.): PKDD 2007, LNAI 4702, pp. 127–139, 2007.

are online social network services (SNS), weblogs (e.g., [1]), and social bookmarks (e.g., [7]). As the world becomes increasingly interconnected as a "global village"[18], network data have multiplied. For that reason, among others, the needs of mining social network data are increasing. A notable feature of social network data is that it is a particular type of relational data in which the target objects are (in most cases) of a single type, and relations are defined between two objects of the type. Sometimes a social network consists of two types of objects: a network is called an affiliation network or a two-mode network.

Social networks have traditionally been analyzed in the field of social network analysis (SNA) in sociology [16,14]. Popular modes of analysis include centrality analysis, role analysis, and clique and cluster analyses. These analyses produce indices for a network, a node, or sometimes for an edge, that have been revealed as effective for many real-world social networks over the half-century history of social studies. In complex network studies [17,3], which is a much younger field, analysis and modeling of scale-free and small world networks have been conducted. Commonly used features of a network are clustering coefficients, characteristic path lengths, and degree distributions.

Numerous works in the data mining community have analyzed social networks [2,13]. For example, L. Backstrom et al. analyzed the social groups and community structure on LiveJournal and DBLP data [2]. They build eight community features and six individual features, and subsequently report that one feature is unexpectedly effective: for moderate values of k, an individual with k friends in a group is significantly more likely to join if these k friends are themselves mutual friends than if they are not. Apparently, greater potential exists for such new features using a network structure, which is the motivation of this research. Although several studies have been done to identify which features are useful to classify entities, no comprehensive research has been undertaken so far to generate the features effectively, including those used in social studies.

In this paper, we propose an algorithm to generate the various network features that are well studied in social network analysis. We define primitive operators for feature generation to create structural features. The combinations of operators enable us to generate various features automatically, some of which correspond to well-known social network indices (such as centrality measures). By conducting experiments on two datasets, the Cora dataset and @cosme dataset, we evaluate our algorithm.

The contributions of the paper are summarized as follows:

- Our research is intended to bridge a gap between the data mining community and the social science community; by applying a set of operators, we can effectively generate features that are commonly used in social studies.
- The research addresses link-based classification from a novel approach. Because some features are considered as novel and useful, the finding might be incorporated into future studies for improving performance for link-based classification.
- Our algorithm is applicable to social networks (or one-mode networks). Because of the increasing amount of attention devoted to social network data,

especially on the Web, our algorithm can support further analysis of the network data, in addition to effective services such as recommendations of communities.

This paper is organized as follows. Section 2 presents related works of this study. In Section 3, we show details of the indices of social network analysis. In Section 4, we propose our method for feature generation by defining nodesets, operators, and aggregation methods. Section 5 describes experimental results for two datasets, followed by discussion and conclusions.

2 Related Work

Various models have been developed for relational learning. A notable study is that of *Probabilistic Relational Models (PRMs)* [5]. Such models provide a language for describing statistical models over relational schema in a database. They extend the Bayesian network representation to enable incorporation of a much richer relational structure and are applicable to a variety of situations. However, the process of feature generation is decoupled from that of feature selection and is often performed manually. Alexandrin et al. [11] propose a method of statistical relational learning (SRL) with a process for systematic generation of features from relational data. They formulated the feature generation process as a search in the space of a relational database. They apply it to relational data from Citeseer, including the citation graph, authorship, and publication, in order to predict the citation link, and show the usefulness of their method.

C. Perlich et al. [10] also propose aggregation methods in relational data. They present the hierarchy of relational concepts of increasing complexity, using relational schema characteristics and introduce target-dependent aggregation operators. They evaluate this method on the noisy business domain, or IPO domain. They predict whether an offer was made on the NASDAQ exchange and draw conclusions about the applicability and performance of the aggregation operators.

L. Backstrom et al. [2] analyzes community evolution, and shows that some *structural features* characterizing individuals' positions in the network are influential, as well as some *group features* such as the level of activity among members. They apply a decision-tree approach to LiveJournal data and DBLP data, which revealed that the probability of joining a group depends in subtle but intuitively natural ways not just on the number of friends one has, but also on the ways in which they are mutually related. Because of the relevance to our study, we explain the individuals' features used in their research in Table 1; they use eight community features and six individual features. Our purpose of this research can be regarded as generating such features automatically and comprehensively to the greatest degree possible.

Our task is categorized into link-based object classification in the context of link mining. Various methods have been used to address tasks such as loopy belief propagation and mean field relaxation labeling [15]. Although these models are useful and effective, we do not attempt to generate such probabilistic or

Table 1. Features used in [2]

Features related to an individual u and her set S of friends in community C
Number of friends in community ($\|S\|$).
Number of adjacent pairs in $S(\|(u,v)\|u,v \in S \wedge (u,v) \in E_C\|)$.
Number of pairs in S connected via a path in E_C.
Average distance between friends connected via a path in E_C.
Number of community members reachable from S using edges in E_C.
Average distance from S to reachable community members using edges in E_C.

statistical models in this study because it is difficult to compose such models using these basic operations.

3 Social Network Features

In this section, we overview commonly-used indices in social network analysis and complex network studies. We call such attributes *social network features* throughout the paper.

One of the simplest features of a network is its *density*. It describes the general level of linkage among the network nodes. The graph density is defined as the number of edges in a (sub-)graph, expressed as a proportion of the maximum possible number of edges.

Within social network analysis, the centrality measures are an extremely popular index of a node. They measure the structural importance of a node, for example, the power of individual actors. There are several kinds of centrality measures [4]; the most popular ones are as follows:

Degree. The degree of a node is the number of links to others. Actors who have more ties to other actors might be advantaged positions. It is defined as $C_i^D = \frac{k_i}{N-1}$, where k_i is the degree of node i and N is the number of nodes.

Closeness. Closeness centrality emphasizes the distance of an actor to all others in the network by focusing on the distance from each actor to all others. It is defined as $C_i^C = (L_i)^{-1} = \frac{N-1}{\sum_{j \in G} d_{ij}}$, where L_i is the average geodesic distance of node i, and d_{ij} is the distance between nodes i and j.

Betweenness. Betweenness centrality views an actor as being in a favored position to the extent that the actor falls on the geodesic paths between other pairs of actors in the network. It measures the number of all the shortest paths that pass through the node. It is defined as $C_i^B = \frac{\sum_{j<k \in G} n_{jk}(i)/n_{jk}}{(N-1)(N-2)}$, where n_{jk} denotes the number of the shortest paths between nodes j and k, and $n_{jk}(i)$ is the number of those running through node i.

A popular variation of centrality measure is the eigenvector centrality (also known as PageRank or stationary probability). Because we do not target the eigenvector centrality in this paper, we do not explain it here but we will discuss it in Section 6.

Another useful set of network indices is the characteristic path length (sometimes denoted as L) and clustering coefficient (denoted as C), which are the most important and frequently-invoked characteristics of complex network studies.

Characteristic path length. The characteristic path length L is the average distance between any two nodes in the network (or a component).

Clustering coefficient. The clustering for a node is the proportion of edges between the nodes within its neighborhood divided by the number of edges that could possibly exist between them. The clustering coefficient C is the average of clustering of each node in the network.

There are other groups of indices such as structural equivalence (defined on a pair of nodes), and structural holes (defined on a node). We do not explain all the indices but readers can consult literature on social network analysis [16,14].

4 Methodology

In this section, we define the elaborate operators that generate social-network features. Using our model, we attempt to generate features that are often used in social science. Our intuition is simple; recognizing that traditional studies in social science have shown the usefulness of several indices, we can assume that feature generation toward the indices is also useful.

Then, how can we design the operators so that they can effectively construct various types of social network features? Through trial and error, we can come up with the feature generation in three steps; we first select a set of nodes. Then the operators are applied to the set of nodes to produce a list of values. Finally, the values are aggregated into a single feature value. Eventually, we can construct indices such as characteristic path length L, clustering coefficient C, and centralities. Below, we explain each step in detail.

4.1 Defining a Node Set

First, we define a node set. We consider two types of node sets: one is based on a network structure; the other is based on the category of a node.

Distance-based node set. Most straightforwardly, we can choose the nodes that are adjacent to node x. The nodes are, in other words, those of distance one from node x. The nodes with distance two, three, and so on can be defined as well. We define a set of nodes as follows.

- $C_x^{(k)}$: a set of nodes within distance k from x.

Note that $C_x^{(k)}$ does not include node x itself. $C_x^{(\infty)}$ means a set of nodes that are reachable from node x.

Table 2. Operator list

Notation	Input	Output	Description	Stage
$C_x^{(1)}$	node x	a nodeset	adjacent nodes to x	1
$C_x^{(\infty)}$	node x	a nodeset	reachable nodes from x	2
$N_p \cap C_x^{(1)}$	node x	a nodeset	all positive nodes adjacent to x	3
$N_p \cap C_x^{(\infty)}$	node x	a nodeset	all positive nodes reachable from x	3
$s^{(1)}$	a nodeset	a list of values	1 if connected, 0 otherwise	1
t	a nodeset	a list of values	distance between a pair of nodes	1
t_x	a nodeset	a list of values	distance between node x and other nodes	2
u_x	a nodeset	a list of values	1 if the shortest path includes node x, 0 otherwise	2
Avg	a list of values	a value	average of values	1
Sum	a list of values	a value	summation of values	1
Min	a list of values	a value	minimum of values	1
Max	a list of values	a value	maximum of values	1
$Ratio_p$	two values	value	ratio of value on positive nodes($N_p \cap C_x^{(k)}$) by all nodes ($C_x^{(k)}$)	4

Category-based node set. We can define a set of nodes with a particular value of some attribute. Although various attributes can be targeted, for link-based classification, we specifically examine the value of the category attribute of a node to be classified. We denote a set of positive nodes as N_p.

Considering both distance-based and category-based node sets, we can define the conjunction of the sets, e.g., $C_x^{(1)} \cap N_p$.

4.2 Operation on a Node Set

Given a nodeset, we can conduct several calculations to the node set. Below, we define operators to two nodes, and then expand it to a nodeset with an arbitrary number of nodes.

The most straightforward operation for two nodes is to check whether the two nodes are adjacent or not. A slight expansion is performed to check whether the two nodes are within distance k or not. Therefore, we define the operator as follows:

$$s^{(k)}(x, y) = \begin{cases} 1 & \textit{if nodes } x \textit{ and } y \textit{ are connected within } k \\ 0 & \textit{otherwise} \end{cases}$$

Another simple operation for two nodes is to measure the geodesic distance between the two nodes on the graph. We can define an operator as follows:

$$t(x, y) = \text{distance between } x \text{ to } y = \arg\min_k \{s^{(k)}(x, y) = 1\}$$

If given a set of more than two nodes (denoted as N), these two operations are applied to each pair of nodes in N. For example, if we are given a node set

$\{n_1, n_2, n_3\}$, we calculate $s^{(1)}(n_1, n_2)$, $s^{(1)}(n_1, n_3)$, and $s^{(1)}(n_2, n_2)$ and return a list of three values, e.g. $(1, 0, 1)$. We denote this operation as $s^{(1)} \circ N$.

In addition to s and t operations, we define two other operations. One is to measure the distance from node x to each node, denoted as t_x. Instead of measuring the distance of two nodes, $t_x \circ N$ measures the distance of each node in N from node x. Another operation is to check the shortest path between two nodes. Operator $u_x(y, z)$ returns 1 if the shortest path between y and z includes node x. Consequently, $u_x \circ N$ returns a set of values for each pair of $y \in N$ and $z \in N$. Operations t_x and u_x focus on node x in terms of the distance and the shortest path, and can be considered fundamental.

4.3 Aggregation of Values

Once we obtain a list of values, several standard operations can be added to the list. Given a list of values, we can take the summation (Sum), average (Avg), maximum (Max), and minimum (Min). For example, if we apply Sum aggregation to a value list $(1, 0, 1)$, we obtain a value of 2. We can write the aggregation as e.g., $Sum \circ s^{(1)} \circ N$. Although other operations can be performed, such as taking the variance or taking the mean, we limit the operations to the four described above.

Additionally, we can take the difference or the ratio of two obtained values. For example, if we obtain 2 by $Sum \circ s^{(1)} \circ N$ and 1 by $Sum \circ s^{(1)} \circ C_x$, the ratio is $2/1 = 2.0$.

We can thereby generate a feature by subsequently defining a nodeset, applying an operator, and aggregating the values. Because the number of possible combinations is enormous, we apply some constraints on the combinations. First, when defining a nodeset, k is an arbitrary integer theoretically; however, we limit k to be 1 or infinity for simplicity. Operator $s^{(k)}$ is used only as $s^{(1)}$. We also limit taking the ratio only to those two values with and without a positive nodeset.

The nodesets, operators, and aggregations are shown in Table 2. We have $4(nodesets) \times 4(operators) \times 4(aggregations) = 64$ combinations. If we consider the ratio, there are ratios for $C_x^{(1)}$ to $N_p \cap C_x^{(1)}$, and for $C_x^{(\infty)}$ to $N_p \cap C_x^{(\infty)}$. In all, there are $4 \times 4 \times 2$ more combinations, and 96 in total. Each combination corresponds to a feature of node x. Note that some combinations produce the same value; for example, $Sum \circ t_x \circ C_x^{(1)}$ is the same as $Sum \circ s \circ C_x^{\infty}$, representing the degree of node x.

The resultant value sometimes corresponds to a well-known index as we intend in the design of the operators. For example, the network density can be denoted as $Avg \circ s^{(1)} \circ N$. It represents the average of edge existence among all nodes; it therefore corresponds to the density of the network. Below, we describe other examples that are used in the social network analysis literature.

- diameter of the network: $Min \circ t \circ N$
- characteristic path length: $Avg \circ t \circ N$
- degree centrality: $Sum \circ s_x^{(1)} \circ N_x^{(1)}$
- node clustering: $Avg \circ s^{(1)} \circ N_x^{(1)}$

- closeness centrality: $Avg \circ t_x \circ C_x^{(\infty)}$
- betweenness centrality: $Sum \circ u_x \circ C_x^{(\infty)}$,
- structural holes: $Avg \circ t \circ N_x^{(1)}$

We can generate several features that have been shown to be effective in existing studies [2]. A couple of examples are the following

- Number of friends in community = $Sum \circ S_x^{(1)} \circ (C_x^{(1)} \cap N_p)$ and
- Number of adjacent pairs = $Sum \circ s^{(1)} \circ (N_x^{(1)} \cap N_p)$.

These features represent some of the possible combinations. Some lesser-known features might actually be effective.

5 Experimental Result

In this section, we describe empirical results obtained using our social network feature generation. Through the experiment, we show the usefulness of the generated features toward link-based classification problems. We classify a node into categories using the relations around the node.

5.1 Datasets and Task

After generating features, we investigate which features are better to classify the entities. We employ a decision tree technique following [2] to generate the decision tree (using C4.5 algorithm [12]). We use two datasets: Cora database and @cosme. We first explain the characteristics of these datasets, and then describe the results and findings.

Cora dataset. This dataset, contributed by A. McCallum [9], contains 300,000 scientific papers related to computer science classified into 69 research areas. About 10,000 papers include detailed information about properties such as the title, author names, a journal name, and the year of publication. In addition, each paper has information about its cited literature. We therefore have a citation network in which a node is a paper and an (undirected) edge is a citation. We do not use direction information on edges.

Training and test data are created as follows: we randomly select nodes from among those in the target category and those which cite or are cited by a paper in the target category. We randomly select one-fifth of the whole 69 categories as target categories. For example, in the case of the category *Neural networks in Machine Learning in Artificial Intelligence*, the number of all nodes is 1682; the number of positive nodes (in this category) is 781. Because the negative examples are the nodes which are not in the category but which have a direct relation with other nodes in the category, the settings are more difficult than those used when we select negative examples randomly.

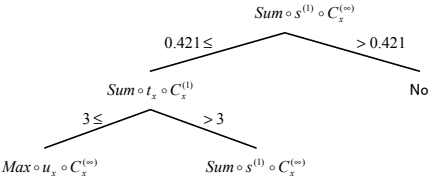

Fig. 1. Top three levels of the decision tree in using up to Stage 2 operators

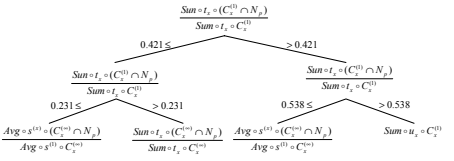

Fig. 2. Top three levels of the decision tree using all operators

@cosme dataset. @cosme (www.cosme.net) is the largest online community site of "for-women" communities in Japan. It provides information and reviews related to cosmetic products. Users of @cosme can post their reviews of cosmetic products (100.5 thousand items of 11 thousand brands) on the system. Notable characteristics of @cosme are that a user can register other users who can be trusted, thereby creating a social network of users.

Because a user of @cosme can join various communities on the site, we can classify users into communities, as was done with the Cora dataset. The nodes are selected from among those who are the members of the community, or those who have a relation with a user in the community. Here we target popular communities with more than 1000 members[1]. In case of *I love Skin Care* communities, the number of nodes is 5730 and the number of positive nodes is 2807.

5.2 Experimental Results

We generate features defined in Table 2 for each dataset. To record the effectiveness of operators, we first limit the operators of Stage 1, as shown in Table 2; then we include the operators of Stage 2, those of Stage 3, and one of Stage 4.

Table 4 shows the values of recall, precision, and F-value for the Cora dataset. The performance is measured by 10-fold cross validation. As we use more operators, the performance improves. Figures 1 and 2 show the top three levels of the decision tree when using operators of Stage 1 and 2, and all the operators. We can see in Fig. 1 that the top level node of the decision tree is $Sum \circ s^{(1)} \circ C_x^{(\infty)}$, which is the number of edges that node x has, or the degree centrality. The second top node is $Sum \circ t_x \circ C_x^{(1)}$, which also corresponds to a degree centrality (in a different expression).

If we add operators in Stage 3 and Stage 4, we obtain a different decision tree as in Fig. 2. The top node is the ratio of the number of positive and all nodes neighboring node x. It means that if the number of neighboring nodes in the category is larger, the node is more likely to be in the category, which can be reasonably understood. We can see in the third level the ratio of $Avg \circ s^{(1)} \circ C_x^{(\infty)}$, which corresponds to the density of the subgraph including node x. There are

[1] Such as *I love Skin Care* community, *Blue Base* community and *I love LUSH* community.

Table 3. Recall, precision, and F-value in the @cosme dataset as adding operators

	Recall	Precision	F-value
Stage 1	0.429	0.586	0.494
Stage 2	0.469	0.593	0.523
Stage 3	0.526	0.666	0.586
Stage 4	0.609	0.668	0.636

Table 4. Recall, precision, and F-value in Cora dataset as adding operators

	Recall	Precision	F-value
Stage 1	0.427	0.620	0.503
Stage 2	0.560	0.582	0.576
Stage 3	0.724	0.696	0.709
Stage 4	0.767	0.743	0.754

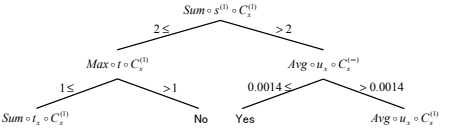

Fig. 3. Top three levels of the decision tree using up to Stage 2 operators in @cosme dataset

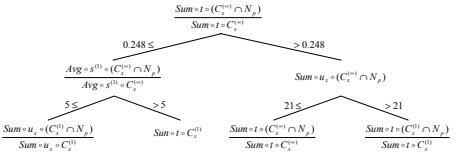

Fig. 4. Top three levels of the decision tree using all operators in @cosme dataset

also features calculating the ratio of $Sum \circ t_x \circ C_x^{\infty}$, which is a closeness centrality, and $Sum \circ u_x \circ C_x^{(1)}$, which corresponds to a betweenness centrality.

The results of @cosme dataset are shown in Table. 3. The trend is the same as that for the Cora dataset; if we use more operators, the performance improves. The decision trees when using up to Stage 2 operators and all operators are shown in Figs. 3 and 4. The top level node of Fig. 3 is $Sum \circ t_x \circ C_x^{(1)}$, which is the number of edges among nodes adjacent to node x. The top level node in Fig. 4 is the ratio of the summation of the path length of reachable positive nodes from node x to the summation of the path length of all reachable nodes. In the third level, we can find $Sum \circ t \circ C_x^{(1)}$. This value is not well known in social network analysis, but it measures the distance among neighboring nodes of node x. The distance is 1 if the nodes are connected directly, and 2 if the nodes are not directly connected (because the nodes are connected via node x). Therefore, it is similar to clustering of node x. Table 5 shows the effective combinations of operators (which appear often in the obtained decision trees) in Cora dataset[2].

In summary, various features have been shown to be important for classification, some of which correspond to well-known indices in social network analysis such as degree centrality, closeness centrality, and betweenness centrality. Some indices seem new, but their meanings resemble those of the existing indices. Nevertheless, the ratio of values on positive nodes to all nodes is useful in many cases. The results support the usefulness of the indices that are commonly used in the

[2] The score $1/r$ is added to the combination if it appears in the r-th level of the decision tree, and we sum up the scores in all the case. (Though other feature weighting is possible, we maximize the correspondence to the decision trees explained in the paper.)

Table 5. Effective combinations of operators in Cora dataset

Rank	Combination	Description
1	$Sum \circ t_x \circ (C_x^{(1)} \cap N_p)$	The number of positive nodes adjacent to node x.
2	$Sum \circ t_x \circ C_x^{(1)}$	The number of nodes adjacent to node x.
3	$Sum \circ s^{(1)} \circ (C_x^{(\infty)} \cap N_p)$	The density of the positive nodes reachable from node x.
4	$Sum \circ s^{(1)} \circ (C_x^{(1)} \cap N_p)$	The number of edges among positive nodes adjacent to node x.
5	$Max \circ t \circ (C_x^{(1)} \cap N_p)$	Whether there is a triad including node x and two positive nodes.
6	$Sum \circ s^{(1)} \circ C_x^{(1)}$	The number of edges among nodes adjacent to node x.
7	$Sum \circ s^{(1)} \circ C_x^{(\infty)}$	The number of edges among nodes reachable from node x.
8	$Max \circ u_x \circ (C_x^{(\infty)} \cap N_p)$	Whether the shortest path includes node x.
9	$Max \circ s^{(1)} \circ (C_x^{(1)} \cap N_p)$	Whether there is a triad including node x and two positive nodes.
10	$Ave \circ s^1 \circ C_x^{(\infty)}$	The Density of the component.

social network literature, and illustrate the potential for further composition of useful features.

6 Discussion

We have determined the operators so that they remain simple but cover a variety of indices. There are other features that can not be composed in our current setting, but which are potentially composable. Examples include

- centralization: e.g., $Max_{n \in N} \circ Sum \circ s^{(1)} \circ C_x^{(\infty)} - Avg_{n \in N} \circ Sum \circ s^{(1)} \circ C_x^{(\infty)}$
- clustering coefficient: $Avg_{n \in N} \circ Avg \circ s^{(1)} \circ N$,

both need additional operators. There are many other operators; for example, we can define the distance of two nodes according to the probability of attracting a random surfer. Eigenvector centrality is a difficult index to implement using operators because it requires iterative processing (or matrix processing). We do not argue that the operators that we define are optimal or better than any other set of operators; we show the first attempt for composing network indices. Elaborate analysis of possible operators is an important future task.

One future study will compare the performance with other existing algorithms for link-based classification, i.e., *approximate collective classification algorithms* (ACCA) [15]. Our algorithm falls into a family of models proposed in Inductive Logic Programming (ILP) called *propositionalization* and *upgrade*. More detailed discussion of the relations to them is available in a longer version of the paper.

7 Conclusions

In this paper, we proposed an algorithm to generate various network features that are well studied in social network analysis. We define operators to generate the features using combinations, and show that some of which are useful for node classification. Both the Cora dataset and @cosme dataset show similar trends. We can find empirically that commonly-used indices such as centrality measures and density are useful ones among all possible indices. The ratio of values, which has not been well investigated in sociology studies, is also sometimes useful.

Although our analysis is preliminary, we believe that our study shows an important bridge between the KDD research and social science research. We hope that our study will encourage the application of KDD techniques to social sciences, and vice versa.

References

1. Adamic, L., Glance, N.: The political blogosphere and the 2004 u.s. election: Divided they blog. In: LinkKDD-2005 (2005)
2. Backstrom, L., Huttenlocher, D., Lan, X., Kleinberg, J.: Group formation in large social networks: Membership, growth, and evolution. In: Proc. SIGKDD'06 (2006)
3. Barabási, A.-L.: LINKED: The New Science of Networks. Perseus Publishing, Cambridge, MA (2002)
4. Freeman, L.C.: Centrality in social networks: Conceptual clarification. Social Networks 1, 215–239 (1979)
5. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In: Proc. IJCAI-99, pp. 1300–1309 (1999)
6. Getoor, L., Diehl, C.P.: Link mining: A survey. SIGKDD Explorations, 2(7) (2005)
7. Golder, S., Huberman, B.A.: The structure of collaborative tagging systems. Journal of Information Science (2006)
8. Lu, Q., Getoor, L.: Link-based classification using labeled and unlabeled data. In: ICML Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining (2003)
9. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Automating the construction of internet portals with machine learning. Information Retrieval Journal 3, 127–163 (2000), `http://www.research.whizbang.com/data.`
10. Perlich, C., Provost, F.: Aggregation based feature invention and relational concept classes. In: Proc. KDD 2003 (2003)
11. Popescul, A., Ungar, L.: Statistical relational learning for link prediction. In: IJCAI03 Workshop on Learning Statistical Models from Relational Data (2003)
12. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, California (1993)
13. Sarkar, P., Moore, A.: Dynamic social network analysis using latent space models. SIGKDD Explorations: Special Edition on Link Mining (2005)
14. Scott, J.: Social Network Analysis: A Handbook, 2nd edn. SAGE publications (2000)

15. Sen, P., Getoor, L.: Link-based classification. In: Technical Report CS-TR-4858, University of Maryland (2007)
16. Wasserman, S., Faust, K.: Social network analysis. Methods and Applications. Cambridge University Press, Cambridge (1994)
17. Watts, D.: Six Degrees: The Science of a Connected Age. W. W. Norton & Company (2003)
18. Wellman, B.: The global village: Internet and community. The Arts & Science Review, University of Toronto 1(1), 26–30 (2006)