Unsupervised Link Selection in Networks

Quanquan Gu
Dept. of Computer Science
UIUC
qgu3@illinois.edu

Charu Aggarwal
IBM T.J. Watson Research Center
Yorktown Heights, NY
charu@us.ibm.com

Jiawei Han
Dept. of Computer Science
UIUC
hanj@cs.uiuc.edu

Abstract

Real-world networks are often noisy, and the existing linkage structure may not be reliable. For example, a link which connects nodes from different communities may affect the group assignment of nodes in a negative way. In this paper, we study a new problem called link selection, which can be seen as the network equivalent of the traditional feature selection problem in machine learning. More specifically, we investigate unsupervised link selection as follows: given a network, it selects a subset of informative links from the original network which enhance the quality of community structures. To achieve this goal, we use Ratio Cut size of a network as the quality measure. The resulting link selection approach can be formulated as a semi-definite programming problem. In order to solve it efficiently, we propose a backward elimination algorithm using sequential optimization. Experiments on benchmark network datasets illustrate the effectiveness of our method.

1 Introduction

Advances in web, social and information network technology have led to the ubiquity of network and graph representations in a wide variety of real-life applications. Examples include the web graph connected by hyper-links, social networks connected by friendship links, bibliographic networks connected by collaboration or citation relationships, and gene or protein networks connected by regulatory relationships. The increasingly important role of network data has lead to significant advances in analytical learning methods. In network data analysis, input data from applications are often noisy, erroneous or unreliable. Since linkages may often be generated by individual user actions in social domains, or statistical inference methods in biological domains, many of these links may actually be noisy and erroneous from an *overall* network analysis perspective. For example, a link which spans nodes from widely separated communities may often not be helpful in the clustering process, and should therefore be removed from the network. Therefore, it is desirable to devise methods which are able to remove those harmful links in a network.

This problem brings to mind, the traditional feature selection problem in the literature. However, existing feature selection methods [11] cannot be applied to our problem, because they are inherently designed for the multidimensional case. Moreover, they are not able to exploit the structure information contained in the links, which is crucial for network analysis. In order to address this problem, we propose to study a new problem known as *Link Selection* for learning in networks. To some extent, link selection can be seen as the network equivalent of feature selection for i.i.d. data.

In this paper, we will design a method for *unsupervised* link selection in networks, based purely on the link structure of a network. The major challenge in this scenario is that no supervision (from humans or labels) may be available for the selection process. We refer to this problem as *unsupervised* link selection, which is analogous to the unsupervised feature selection problem in machine learning. More specifically, we use the ratio cut size [17] as the criterion, which provides insights about the structural quality of links. This leads to a combinatorial optimization problem, which can be further relaxed into a semi-definite programming (SDP) problem [3]. Nevertheless, the computational complexity of SDP is too high to be practical. Therefore, we derive a sequential optimization algorithm to solve it, where the first order Taylor expansion of the objective function is minimized iteratively. We experi-

Appearing in Proceedings of the 16^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2013, Scottsdale, AZ, USA. Volume 31 of JMLR: W&CP 31. Copyright 2013 by the authors.

mentally demonstrate the effectiveness of the proposed approach on community detection in real networks.

The remainder of this paper is organized as follows. In Section 2, we briefly review some related work. In Section 3, we present a link selection algorithm, which is based purely on linkage information. The experimental results are presented in Section 4. The conclusions and future work are presented in Section 5.

2 Related Work

In this section, we review the related work on general feature selection, community detection methods, and the recent work on feature selection in the network analysis context.

2.1 Feature Selection

Feature selection has been widely recognized as an effective method [11] for improving the data quality for a variety of learning problems. In general, feature selection [22] can be classified into three families, corresponding to *filter-based* [12] [10], *wrapper-based*, and *embedded methods* [11]. Filter-based methods score the features as a pre-processing step, independently of the classifier. Wrapper-based methods score the features according to their prediction performance when used with the classifier. Finally, embedded methods tightly integrate the selection method with the specific classifier. Such methods are often considered more effective than filters and wrappers [11]. The link selection method proposed in this paper shares the same spirit of embedded methods, because it is built upon ratio cut-based spectral graph partitioning [17].

On the other hand, depending whether there are labels, feature selection can be categorized into unsupervised and (semi-)supervised methods. Unsupervised feature selection is attractive because in many application scenarios, labels are often unavailable. Typical unsupervised feature selection methods include Laplacian score [12] and α-Q [20]. The proposed link selection method in this paper is also unsupervised.

We also note that there are several works which study feature selection in networks [9] [16]. Different from link selection, these methods select the attributes of node content, while our work is focussed on the linkage structure.

2.2 Community Detection

Community detection has been widely studied in recent years. From a linkage viewpoint, communities may be considered groups of nodes which are densely connected by edges in the network. In the past decade, many community detection methods have been proposed in the literature. Conventional algorithms [21] [23] use aggregate linkage analysis for the community detection process. These methods typically exploit the topological or statistical correlations in the network linkage structure for community detection. Modularity-based methods [15] [1] measure the strength of community structure in terms of the difference between the expected number of edges in the community from the true number of edges. Spectral clustering [17] is a family of graph partitioning methods, which aim to minimize the cut size (e.g., min cut, ratio cut, or normalized cut) of a network [19] [4]. [14] studied the statistical property of community structures in large networks. There are also some works [13] which connect community (clique) detection with basis pursuit.

3 Unsupervised Link Selection Based on Link Analysis

This section introduces the problem definition and method for unsupervised link selection.

3.1 Problem Definition

For notational clarity, we consistently use lower case letters to denote scalars, lower case bold letters to denote vectors, and bold-face upper case letters to denote matrices. We denote the vector of all zeros by $\mathbf{0}$, and the identity matrix by $\mathbf{I}$.

Given a network $G = (V, E)$ with node set V ($|V| = n$), and edge set E ($|E| = m$), we denote the ith node by $v_i \in V$, and an edge between nodes i and j by $e_{ij} \in E$. The weight of edge $e_{ij} \in E$ is denoted by A_{ij}. Therefore, the adjacency matrix of the graph is denoted by $\mathbf{A} \in \mathbb{R}^{n \times n}$.

The generic problem of link selection in networks is as follows. Given a network G where the cardinality of the edge set E is m, our goal is to find a subset $S \subset E$ of $l < m$ most noisy edges, which should be removed from the network.

Since our criterion for evaluating the quality of links is based on Ratio Cut, we will first introduce the underlying concepts.

3.2 Ratio Cut

For a partitioning of the network into c communities, dented by $C_1, \ldots, C_c$, the *Ratio Cut* of this partition is defined as follows:

$$\text{RatioCut}(C_1, \ldots, C_c) = \sum_{k=1}^{c} \frac{\text{cut}(C_k, \bar{C}_k)}{|C_k|} \tag{1}$$

where $\bar{C}_k$ is the complementary set of C_k, $\text{cut}(C_k, \bar{C}_k) = \sum_{i \in C, j \in \bar{C}_k} A_{ij}$.

We introduce the cluster assignment matrix, denoted by $\mathbf{F} \in \mathbb{R}^{n \times c}$, where $F_{ik} = 1$ if vertex v_i belongs to the k-th group, and $F_{ik} = 0$ otherwise. Therefore, each row of matrix $\mathbf{F}$ contains exactly one 1, and the remaining elements are all zeros. We define a scaled cluster assignment matrix $\mathbf{P} \in \mathbb{R}^{n \times c}$, such that $P_{ik} = F_{ik}/\sqrt{n_k}$ where n_k is the number of vertices in the k-th cluster. Then, we have:

$$\mathbf{P} = \mathbf{F}(\mathbf{F}^\top \mathbf{F})^{-1/2} \tag{2}$$

The scaled cluster assignment matrix is useful for concise expression of the *RatioCut* [17]. We note that the scaled assignment matrix $\mathbf{P}$ is a semi-orthogonal matrix, because we have:

$$\mathbf{P}^\top \mathbf{P} = (\mathbf{F}^\top \mathbf{F})^{-1/2} \mathbf{F}^\top \mathbf{F} (\mathbf{F}^\top \mathbf{F})^{-1/2} = \mathbf{I}. \tag{3}$$

According to [17], the *RatioCut* can be rewritten concisely in terms of $\mathbf{P}$ as follows:

$$\begin{aligned} \text{RatioCut}(\mathbf{F}) &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{c} (P_{ik} - P_{jk})^2 A_{ij} \\ &= \frac{1}{2} \text{tr}(\mathbf{P}^\top \mathbf{L} \mathbf{P}) \end{aligned} \tag{4}$$

where $\mathbf{D}$ is a diagonal matrix, called the *weighted degree matrix*, with $D_{ii} = \sum_{j=1}^{n} A_{ij}$, and $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the combinatorial graph Laplacian [5]. From the graph regularization point of view, it implies that if the i-th node and the j-th node are connected ($A_{ij} > 0$), then their cluster assignments (P_{ik} and P_{jk}) tend to be similar.

The spectral relaxation of the Ratio cut aims at finding $\mathbf{P} \in \mathbb{R}^{n \times c}$, such that the cut size computed based on $\mathbf{P}$ is minimized:

$$\begin{aligned} \arg \min_{\mathbf{P}} \quad & \text{tr}(\mathbf{P}^\top \mathbf{L} \mathbf{P}) \\ \text{s.t.} \quad & \mathbf{P}^\top \mathbf{P} = \mathbf{I}, \end{aligned} \tag{5}$$

The optimal solution of the above problem can be obtained by determining the eigenvectors which correspond to the smallest k eigenvalues.

On the other hand, according to [2], the graph Laplacian $\mathbf{L}$ can be represented by the dot product summation of edge vectors. For an edge e_{ij} connecting two nodes i and j, we define the edge vector $\mathbf{e}_{ij} \in \mathbb{R}^n$ where $\mathbf{e}_{ij} = [\ldots, 1, \ldots, -1, \ldots]^\top$ where the i-th element is 1, and the j-th element is -1. Suppose there are m edges in the graph G, then the graph Laplacian $\mathbf{L}$ can be written as follows:

$$\mathbf{L} = \sum_{e_{ij} \in E} A_{ij} \mathbf{e}_{ij} \mathbf{e}_{ij}^\top \tag{6}$$

For each edge $e_{ij} \in E$, we introduce a binary selection variable $s_{ij} \in \{0, 1\}$. If e_{ij} is selected for removal, then we have $s_{ij} = 1$, and otherwise we have $s_{ij} = 0$. Then the graph Laplacian $\mathbf{L}$ can be written as follows:

$$\mathbf{L} = \sum_{e_{ij} \in E} (1 - s_{ij}) A_{ij} \mathbf{e}_{ij} \mathbf{e}_{ij}^\top \tag{7}$$

Next, we will model the problem of optimizing link selection.

3.3 Objective Function

The basic idea of our proposed link selection is as follows. We would like to remove edges, such that the minimum cut applied on the new network is minimized. This problem is mathematically formulated as follows,

$$\begin{aligned} \arg \min_{s_{ij} \in \{0,1\}} \min_{\mathbf{P}^\top \mathbf{P} = \mathbf{I}} \quad & \text{tr}(\mathbf{P}^\top \mathbf{L} \mathbf{P}) \\ \text{s.t.} \quad & \mathbf{L} = \sum_{e_{ij} \in E} (1 - s_{ij}) A_{ij} \mathbf{e}_{ij} \mathbf{e}_{ij}^\top \\ & \sum_{e_{ij} \in E} s_{ij} = l \end{aligned} \tag{8}$$

In the above equation, the inner minimization is a standard spectral clustering process corresponding to the minimization of the ratio cut over a specific selection configuration, while the outer minimization is over all possible selection configurations.

It is well known that the optimal value of the inner minimization corresponds to the sum of top-c smallest eigenvalues of the graph Laplacian $\mathbf{L}$. Therefore, the above min-min problem can be simplified as single minimization problem as follows

$$\begin{aligned} \arg \min_{s_{ij} \in \{0,1\}} \quad & \sum_{k=1}^{c} \lambda_k(\mathbf{L}) \\ \text{s.t.} \quad & \mathbf{L} = \sum_{e_{ij} \in E} (1 - s_{ij}) A_{ij} \mathbf{e}_{ij} \mathbf{e}_{ij}^\top \\ & \sum_{e_{ij} \in E} s_{ij} = l \end{aligned} \tag{9}$$

where $\lambda_k(\mathbf{L})$ is the k-th smallest eigenvalue of $\mathbf{L}$.

The above optimization is a combinatorial optimization problem over a domain of integer solutions. One way to solve it is to relax s_{ij} into the continuous domain, i.e., $s_{ij} \in [0, 1]$. Then the above problem becomes a semi-definite programming (SDP) [3] [2] problem. However, even with recent advances in interior point methods, solving a large scale SDP is computationally prohibitive. Therefore, in the following, we will present a sequential optimization algorithm, which is much simpler and more efficient.

3.4 Sequential Optimization

In the process of sequential optimization, we remove one link at a time. When t links have been removed, we denote the Laplacian of the remaining graph by $\mathbf{L}_t$. Therefore, we have $\mathbf{L}_0 = \mathbf{L}$. And it is easy to show that, if edge e_{ij} is removed in the $(t+1)$-th step, we have $\mathbf{L}_{t+1} = \mathbf{L}_t - s_{ij}A_{ij}\mathbf{e}_{ij}\mathbf{e}_{ij}^\top$ by simple linear algebraic operation. Then, the $(t+1)$-th link can be selected for removal by solving the following optimization problem:

$$\begin{aligned}\arg\min_{s_{ij}\in\{0,1\}} \quad & \sum_{k=1}^{c}\lambda_k(\mathbf{L}_{t+1}) \\ = \quad & \sum_{k=1}^{c}\lambda_k(\mathbf{L}_t - \sum_{e_{ij}\in E_t} s_{ij}A_{ij}\mathbf{e}_{ij}\mathbf{e}_{ij}^\top) \\ \text{s.t.} \quad & s_{ij}\in\{0,1\}, \sum_{e_{ij}\in E_t} s_{ij} = 1 \end{aligned} \tag{10}$$

where $E_t \subset E$ is the remaining edge set after t links are removed.

The afore-mentioned objective function can be optimized with the use of the first-order Taylor expansion at $\mathbf{L}_t$. Recall that

$$\lambda_k(\mathbf{L})\mathbf{v}_k = \mathbf{L}\mathbf{v}_k \tag{11}$$

where $\mathbf{v}_k$ is the eigenvector corresponding to the k-th smallest eigenvalue of $\mathbf{L}$, i.e., $\lambda_k(\mathbf{L})$. We pre-multiply the vector $\mathbf{v}_k$ to both sides of Eq. (11) in order to obtain the following:

$$\mathbf{v}_k^\top\lambda_k(\mathbf{L})\mathbf{v}_k = \mathbf{v}_k^\top\mathbf{L}\mathbf{v}_k \tag{12}$$

Because $\mathbf{v}_k$ is normalized, i.e., $\|\mathbf{v}_k\|_2 = 1$, we have

$$\lambda_k(\mathbf{L}) = \mathbf{v}_k^\top\mathbf{L}\mathbf{v}_k \tag{13}$$

Therefore, the first-order partial derivative of $\lambda_k(\mathbf{L}_{t+1})$ with respect to s_{ij} at $s_{ij}=0$ is as follows:

$$\begin{aligned} & \left.\frac{\partial\lambda_k(\mathbf{L}_{t+1})}{\partial s_{ij}}\right|_{s_{ij}=0} \\ = \quad & \left.\mathbf{v}_k^\top\frac{\partial\mathbf{L}_{t+1}}{\partial s_{ij}}\mathbf{v}_k\right|_{s_{ij}=0} \\ = \quad & \left.\mathbf{v}_k^\top\frac{\partial(\mathbf{L}_t - \sum_{e_{ij}\in E_t} s_{ij}A_{ij}\mathbf{e}_{ij}\mathbf{e}_{ij}^\top)}{\partial s_{ij}}\mathbf{v}_k\right|_{s_{ij}=0} \\ = \quad & -\mathbf{v}_k^\top(A_{ij}\mathbf{e}_{ij}\mathbf{e}_{ij}^\top)\mathbf{v}_k \\ = \quad & -A_{ij}(\mathbf{v}_k^\top\mathbf{e}_{ij})(\mathbf{e}_{ij}^\top\mathbf{v}_k) \\ = \quad & -A_{ij}(v_{ik}-v_{jk})^2 \end{aligned} \tag{14}$$

where v_{ik} is the i-th element of $\mathbf{v}_k$.

The first-order Taylor expansion of the objective function in Eq. (10) at $s_{ij}=0$ is

$$\begin{aligned} & \sum_{k=1}^{c}\lambda_k(\mathbf{L}_{t+1}) \\ \approx \quad & \sum_{k=1}^{c}\lambda_k(\mathbf{L}_t) - \sum_{k=1}^{c}\sum_{e_{ij}\in E_t} s_{ij}A_{ij}(v_{ik}-v_{jk})^2 \end{aligned} \tag{15}$$

Therefore, the optimization problem in Eq. (10) can be approximately solved by

$$\begin{aligned}\arg\max_{s_{ij}\in\{0,1\}} \quad & \sum_{k=1}^{c}\sum_{e_{ij}\in E_t} s_{ij}A_{ij}(v_{ik}-v_{jk})^2 \\ \text{s.t.} \quad & s_{ij}\in\{0,1\}, \sum_{e_{ij}\in E_t} s_{ij} = 1 \end{aligned} \tag{16}$$

We omit $\sum_{k=1}^{c}\lambda_k(\mathbf{L}_t)$ because it is a constant. The above problem can be solved by sorting $\sum_{k=1}^{c}A_{ij}(v_{ik}-v_{jk})^2$, and set the s_{ij} corresponding to the largest $\sum_{k=1}^{c}A_{ij}(v_{ik}-v_{jk})^2$ to 1.

Once the $(t+1)$-th link is removed, $\mathbf{L}_{t+1}$ can be updated based on $\mathbf{L}_t$, by using the same approach. Therefore, this process is efficient.

In summary, we present the entire algorithmic framework for link selection in networks in Algorithm 1.

Algorithm 1 Link Selection in Networks (**LS**)

```
Input: Adjacency matrix $\mathbf{A}$, number of nodes to remove $l$;
Compute $\mathbf{L} = \mathbf{D} - \mathbf{A}$
Initialize $\mathbf{L}_0 = \mathbf{L}$, $E_0 = E$
for $t = 0 \to l-1$ do
    Compute the eigenvectors $\mathbf{v}_k, k = 1, \ldots, c$ corresponding to the smallest $c$ eigenvalues of $\mathbf{L}_t$;
    Compute $e_{i^*j^*} = \arg\max_{e_{ij}\subset E_t}\sum_{k=1}^{c}A_{ij}(v_{ik}-v_{jk})^2$;
    Update $E_{t+1} = E_t \setminus \{e_{i^*j^*}\}$
    Update $\mathbf{L}_{t+1} = \mathbf{L}_t - A_{i^*j^*}\mathbf{e}_{i^*j^*}\mathbf{e}_{i^*j^*}^T$
end for
```

It is worth noting that Algorithm 1 monotonically decreases the objective function value in Eq.(9), which is a upper bound of the ratio cut size [5]. This property is appealing because it may provide a potential way to choose the number of links to remove (model selection).

3.5 Complexity Analysis

In each iteration of the algorithm, it requires $O(n^2c)$ time to calculate the top-c eigenvectors using Lanczos algorithm [7]. It takes $O(m\cdot\log_2(m))$ to perform sorting. Therefore, the total complexity of the algorithm is $O(l(n^2c + m \cdot \log_2(m)))$. In practice, the complexity can be further reduced because we use the eigenvectors obtained in last round as the initialization for the Lanczos in current round, which leads to considerable speedup.

3.6 Discussion

Here we would like to give an intuitive interpretation of our algorithm. Recall that typical spectral clustering methods [17] usually consist of two steps. The first step is computing the low-dimensional embedding of the graph by eigenvalue decomposition of the (normalized) graph Laplacian. The second step is treating the low-dimensional embedding as a new data space, and applying k-means algorithm on this low-dimensional space. By our notation, $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_c]$ is the low-dimensional embedding of a network, and each row of $\mathbf{V}$ can be seen as a data point in the new space. In Eq. (16), we would like to find an edge such that $\sum_{k=1}^{c} s_{ij} A_{ij} (v_{ik} - v_{jk})^2$ is maximized. From the spectral embedding point of view, we actually find two data points in the low-dimensional space, such that the distance between these two data points is the farthest. Since each data points in the low-dimensional space corresponds to a node in the network, it can be understood that we want to find two connected nodes in the network whose similarity is the smallest, and then disconnect these two nodes.

4 Experiments

In this section, we evaluate the proposed link selection method on both synthetic and real-world network datasets, and investigate their impact on community detection.

4.1 Synthetic Data Case Study

To get an intuitive picture of how our proposed link selection methods work, we generate several very small synthetic datasets which have obvious community structures in Figure 1. For each synthetic data, the node color represents the community membership of each node. Thus, there are two communities in the first synthetic data, and three communities in the second and third datasets. Note that the first synthetic data is the motivating example we have seen in Section 1. Since there is only linkage information in these datasets, we apply the LS (Algorithm 1) to them. For the synthetic data 1, we let the algorithm to find one link to remove. For synthetic data sets 1, 2, and 3, we determine the top one, two and three links respectively to remove. The links which were selected by our method for removal are denoted in green. It is clear, that in each case, our method can detect the noisy links correctly. It is evident that the removal of such noisy links would clean up the structure of the network in order to enable a more effective application of practical community detection methods, without being confused by the anomalous links.

4.2 Real Datasets

We also used four real-world benchmark datasets for evaluation. The nodes were typically labeled on the basis of certain community-centric properties, and this provided useful information for evaluation purposes. These datasets are as follows:

Journals[1]: In this dataset, over 100,000 people were surveyed in 1999-2000, about their preference in magazines and journals (source CATI Center Ljubljana). They listed 124 different magazines and journals. Each node corresponds to a journal (or magazne), and an edge with a value between journals means the number of readers of both journals. The labels are the categories of the journals.

BlogCatalog[2] is a blog directory where users can register their blogs under predefined categories. During the registry process of a new blog, the user is asked to specify the major category and a subcategory in a hierarchical structure, and specify several tags to describe the main topics of the blog. The labels are the group tags of the blogs. This dataset was used in [18].

Coauthor is an undirected co-author graph data extracted from the DBLP[3] database in four areas: machine learning, data mining, information retrieval and database. It contains a total of 1711 authors, each of which is represented by a node. The edge between each pair of authors is weighted by the number of papers they co-authored. Each class contains about 400 authors.

PubMed[4] consists of 19717 scientific publications from PubMed database pertaining to diabetes classified into one of three classes. The citation network consists of 44338 links.

Some statistics of the datasets are shown in Table 1.

4.3 Evaluation Measures

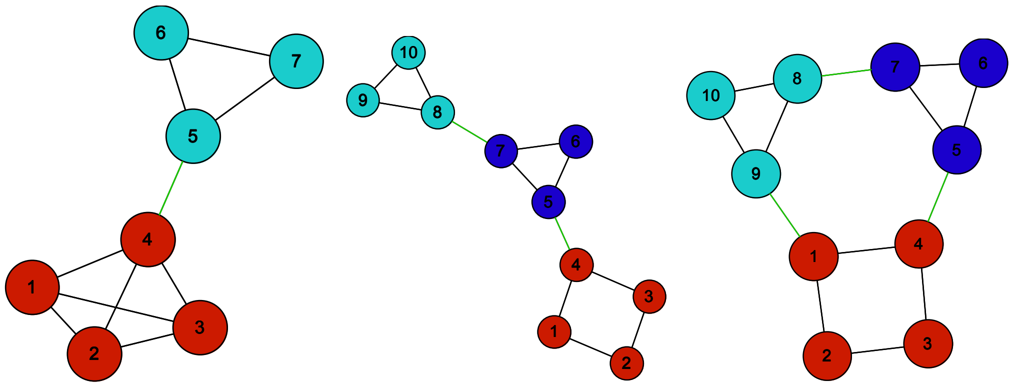
In order to measure the effectiveness of our approach, we adopt two measures to evaluate the quality of the

[1] http://vlado.fmf.uni-lj.si/pub/networks/data/2mode/journals.htm

[2] http://www.blogcatalog.com

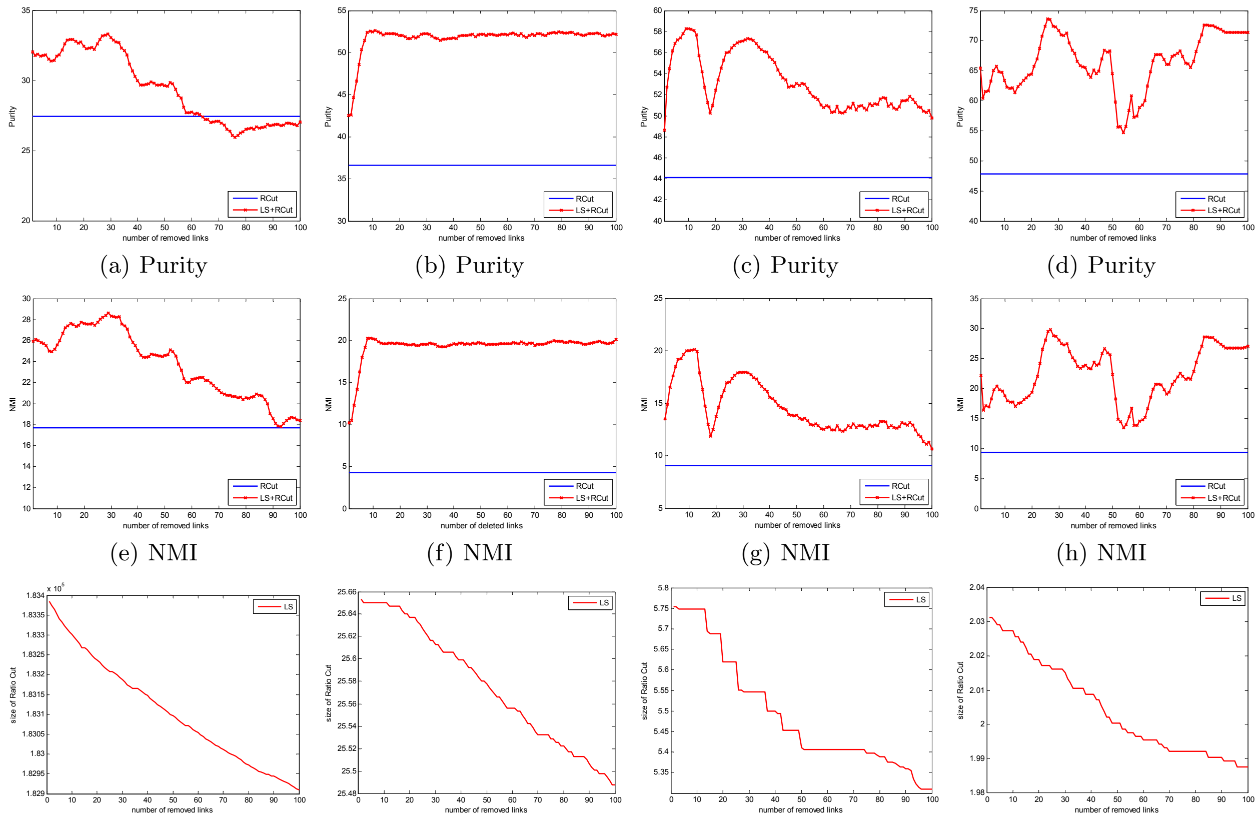
[3] www.informatik.uni-trier.de/∼ley/db/

[4] http://www.cs.umd.edu/projects/linqs/projects/lbc/Pubmed-Diabetes.tgz



(a) Synthetic Data 1 (b) Synthetic Data 2 (c) Synthetic Data 3

Figure 1: Link Selection on three synthetic datasets. The node color represents the community membership of each node. The green edges ((4,5) in (a), (4,5) and (7,8) in (b), (4,5),(7,8),(1,9) in (c)) are the links selected by LS for removal.



(a) Purity (b) Purity (c) Purity (d) Purity

(e) NMI (f) NMI (g) NMI (h) NMI

(i) Ratio Cut Size (j) Ratio Cut Size (k) Ratio Cut Size (l) Ratio Cut Size

Figure 2: A comparison of community detection using Ratio Cut with and without link selection on the first three datasets (link selection only with structure): (1) *Journals* (1st column); (2) *BlogCatalog* (2nd column); (3) *Coauthor* (3rd column) and (4) *Pubmed* (4th column).

communities generated by different approaches. We also use an additional measure to evaluate the correctness of removed links, as discussed below.

Purity The definition of purity is as follows. First, the dominant label of each cluster is determined. The purity is measured by computing the number of the nodes assigned to a cluster with the same dominant

Table 1: The statistics of the network datasets

	#Nodes	#Links	#Communities
Journals	124	12068	14
BlogCatalog	2497	27878	4
Coauthor	4057	32789	4
Pubmed	19717	44338	3

label. Formally:

$$\text{Purity} = \frac{1}{n}\sum_{i=1}^{c}\max_{1\leq j\leq c}|C_i \cap L_j| \tag{17}$$

where $C = \{C_1, \ldots, C_c\}$ is the set of clusters, and $L = \{L_1, \ldots, L_c\}$ is the set of ground truth classes. $|C_i \cap L_j|$ is the intersection between the cluster C_i and the class L_j. The value of purity ranges from 0 to 1. Clearly, higher values of the accuracy are more desirable.

Normalized Mutual Information The second measure is the Normalized Mutual Information (NMI), which is used for determining the quality of clusters. Given a clustering result, the mutual information metric is defined as follows:

$$MI(C, L) = \sum_{i=1}^{c}\sum_{j=1}^{c} p(C_i, L_j)\log_2\frac{p(C_i, L_j)}{p(C_i)p(L_j)}, \tag{18}$$

where $p(C_i)$ and $p(L_j)$ are the probabilities that a node arbitrarily selected from the network belongs to the clusters C_i and L_j, respectively, and $p(C_i, L_j)$ is the joint probability that the arbitrarily selected node belongs to the clusters C_i as well as class L_j at the same time. In our experiments, we use the normalized mutual information as follows:

$$NMI(C, L) = \frac{MI(C, L)}{\max\{H(C), H(L)\}} \tag{19}$$

where $H(C)$ and $H(L)$ are the entropies of C and L, respectively The value of $NMI(C, L)$ ranges from 0 to 1. Larger values of NMI are more desirable.

Ratio Cut Size: In order to evaluate the quality of the network after link selection directly, we also calculate the size of Ratio Cut (as defined in Eq. (1)). Recall that our goal is to remove links such that the ratio cut size of the remained network is decreased.

4.4 Experimental Setup

As the baseline, we applied Ratio Cut on the original network (without link selection) [17] for community detection. Note that Ratio cut is a kind of spectral clustering method. The number of clusters to be the number of communities in the ground truth in both methods. After obtaining the low-dimensional embedding of the network by Ratio cut, we apply k-means to it. Due to the randomness of k-means, we repeat k-means 10 times, and the average results (i.e., Purity and NMI) are reported.

For comparison purposes, we also ran Ratio cut on the new network *after link selection* for community detection. For each dataset, we repeated this process at different levels of link removal. As in the case of the baseline, we repeated k-means 10 times on the embedding obtained by Ratio Cut, and the average results (i.e., Purity and NMI) were reported. We applied the pure link-selection algorithm LS (Algorithm 1) on all the datasets.

4.5 Community Detection using Ratio Cut

The experimental results of link selection for Ratio Cut on the real world data sets are shown in Figures 2. In all plots, the X-axis represents the number of removed links. Note that the X-axis begins with 1 removed link rather than 0 removed link. Each row represents the three performance measures for a data set. Specifically, the Y-axis of the leftmost plot in a row represents the purity, that of the middle plot represents the NMI, and that of the rightmost plot represents the ratio cut for the reduced network.

In general, we can see that the link selection process improves the quality of community detection. It is also always better than the baseline. When the number of removed links is too large (> 50), the performance will degrade on most datasets (except the *Pubmed* dataset). The reason for this is that very aggressive link elimination may sometimes result in loss of information. On the other hand, the ratio cut size of the reduced network by LS decreased monotonically with respect to an increasing number of removed edges. This indicates that our methods worked as expected in terms of network-specific measures of quality.

It is also worth noting that even with one link removed, the community detection performance is dramatically improved. This further justifies the importance of link selection, especially the necessity of removing the harmful links.

In summary, the results show that the link selection process is particularly beneficial when a small number of links are removed. Therefore, we show the results of Ratio cut for community detection after link selection methods averaged over top-50 removed links in Tables 2. For each dataset, the best results are illustrated in bold. It can be observed again that LS outperforms the baseline. In fact, we did paired t-tests between the proposed method and the baseline in the 95% confidence interval. We found that the performance gain achieved by link selection was significant.

Table 2: A comparison of community detection quality using Ratio Cut with and without link selection. The performance of community detection with link selection are averaged over the top-50 removed links.

	Purity		NMI	
	RCut	LS+RCut	RCut	LS+RCut
Journals	27.42±0.00	**31.69±1.34**	17.70±1.24	**26.47±1.63**
BlogCatalog	36.62±1.82	**51.30±2.42**	4.25±0.52	**18.92±2.37**
Coauthor	44.10±1.76	**55.31±2.66**	9.07±2.11	**16.48±2.61**
Pubmed	47.81±2.10	**66.38±4.51**	9.36±3.02	**22.74±4.78**

Table 3: A comparison of community detection using Normalized Cut with and without link selection. The performance of community detection with link selection are averaged over the top-50 removed links.

	Purity		NMI	
	NCut	LS+NCut	NCut	LS+NCut
Journals	47.90±1.27	49.11±0.74	49.54±0.98	50.21±0.56
BlogCatalog	31.48±0.68	34.29±3.30	1.13±1.13	2.08±2.17
Coauthor	50.53±1.59	53.74±2.18	15.35±1.17	15.74±1.79
Pubmed	72.37±0.00	51.85±9.49	28.91±0.00	9.92±7.65

4.6 Community Detection using Normalized Cut

One natural question is how well our method works on other community detection algorithms. Because of space limitations, we only show the results of the Normalized cut for community detection after link selection methods averaged over top-50 removed links in Table 3. We can see that on three out of four datasets, our link selection method is able to improve the performance of normalized cut, though the performance gain is somewhat lower. Thus, the link selection methods proposed in this paper are an effective way to clean the underlying structure of networks, so as to improve their representation quality.

5 Conclusions and Future Work

In this paper, we study the problem of link selection in networks. Given a network, we propose to select a subset of informative links from the original network, which improves the quality of link structure. Our approach uses the cut size for the selection process. We present a backward link selection algorithm using sequential optimization. Experiments on benchmark data sets justify the performance improvement obtained by link selection.

In the future, in order to scale the proposed method to networks with millions of nodes, we would like to adapt Nystrom method [6] or rank-One modification algorithm of the symmetric eigen-problem [8] to accelerate eigenvalue decomposition.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. The work was supported in part by U.S. National Science Foundation grants IIS-0905215, CNS-0931975, the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA), the U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265, and MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC.

References

[1] B. Ball, B. Karrer, and M. E. J. Newman. Efficient and principled method for detecting communities in networks. *Phys. Rev. E*, 84:036103, Sep 2011.

[2] S. Boyd. Convex optimization of graph laplacian eigenvalues. In *in International Congress of Mathematicians*, pages 1311–1319.

[3] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, Cambridge, 2004.

[4] J. Chen, O. R. Zaïane, and R. Goebel. Detecting communities in social networks using max-min modularity. In *SDM*, pages 978–989, 2009.

[5] F. R. K. Chung. *Spectral Graph Theory.* American Mathematical Society, February 1997.

[6] P. Drineas and M. W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

[7] G. H. Golub and C. F. V. Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[8] M. Gu and S. C. Eisenstat. A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem. *SIAM J. Matrix Anal. Appl.*, 15(4):1266–1276, Oct. 1994.

[9] Q. Gu and J. Han. Towards feature selection in network. In *CIKM*, pages 1175–1184, 2011.

[10] Q. Gu, Z. Li, and J. Han. Generalized fisher score for feature selection. In *UAI*, pages 266–273, 2011.

[11] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[12] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *NIPS*, 2005.

[13] X. Jiang, Y. Yao, H. Liu, and L. J. Guibas. Detecting network cliques with radon basis pursuit. *Journal of Machine Learning Research - Proceedings Track*, 22:565–573, 2012.

[14] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In *WWW*, pages 695–704, 2008.

[15] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74:036104, Sep 2006.

[16] J. Tang and H. Liu. Feature selection with linked data in social media. In *SIAM International Conference on Data Mining*, 2012.

[17] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[18] X. Wang, L. Tang, H. Gao, and H. Liu. Discovering overlapping groups in social media. In *ICDM*, pages 569–578, 2010.

[19] S. White and P. Smyth. A spectral clustering approach to finding communities in graph. In *SDM*, 2005.

[20] L. Wolf and A. Shashua. Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *Journal of Machine Learning Research*, 6:1855–1887, 2005.

[21] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. Scan: a structural clustering algorithm for networks. In *KDD*, pages 824–833, 2007.

[22] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, pages 412–420, 1997.

[23] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *PVLDB*, 2(1):718–729, 2009.