Towards Feature Selection in Networks

Quanquan Gu
Department of Computer Science
University of Illinois at Urbana-Champaign
IL, 61801, USA
qgu3@illinois.edu

Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign
IL, 61801, USA
hanj@cs.uiuc.edu

ABSTRACT

Traditional feature selection methods assume that the data are independent and identically distributed (i.i.d.). In real world, tremendous amounts of data are distributed in a network. Existing features selection methods are not suited for networked data because the i.i.d. assumption no longer holds. This motivates us to study feature selection in a network. In this paper, we present a supervised feature selection method based on Laplacian Regularized Least Squares (LapRLS) for networked data. We use linear regression to utilize the content information, and adopt graph regularization to consider the link information. The proposed feature selection method aims at selecting a subset of features such that the empirical error of LapRLS is minimized. The resultant optimization problem is a mixed integer programming, which is difficult to solve. It is relaxed into a $L_{2,1}$-norm constrained LapRLS problem and solved by accelerated proximal gradient descent algorithm. Experiments on benchmark networked data sets show that the proposed feature selection method outperforms traditional feature selection method and the state-of-the-art learning-in-network approaches.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.5.1 [**Pattern Recognition**]: Models

General Terms

Algorithms, Experimentation

Keywords

Feature Selection, Network, Graph Regularization, Laplacian Regularized Least Squares

1. INTRODUCTION

In many applications of data mining, one is often confronted with very high dimensional data. It significantly increases the time and space complexity for processing the data. Moreover, in the presence of many irrelevant and/or redundant features, learning methods tend to over-fit and become less interpretable. One way to resolve this problem is feature selection [23] [9], which reduces the dimensionality by selecting a subset of features from the input feature set.

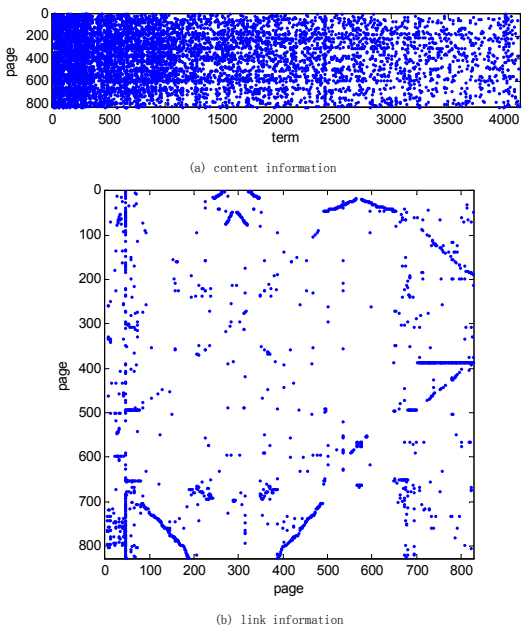


Figure 1: Sample data of WebKB (Cornell) dataset: (a) Content information (page-term co-occurrence matrix); (b) Link information (adjacency matrix). Note that the adjacency matrix is not symmetric because the link has direction.

However, traditional feature selection methods [9] assume that the data are sampled i.i.d. from some unknown distribution. While in real world, there are tremendous amounts of data which are distributed in a network. For example, in the classification of web pages, there are not only the contents within web pages, but also the hyper-links between the web pages. See Figure 1 for example. Figure 1 (a) shows the page-term matrix where a blue dot represents the occurrence of a term in a page. Figure 1 (b) shows the adjacency matrix of the pages, where blue dot represents the link between pages. It is often the case that if there is a link from one page to another, then it is likely that these two pages are related. Another example is research paper classification. Besides

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

the content information of each paper, a citation relation between two papers provides an evidence that they belong to the same topic. That is, if one paper cites another, or is cited by another, then there is a high chance that the papers belong to the same topic. It can be seen that content information and link information complement each other. Hence, if one can utilize both the content information and link information in the network, one should achieve better classification result of the networked data than either using the content information or link information alone. In the past decade, there are quite a lot of studies on learning in network along several directions. For example, combining content and link information for classification [6] [26] [13], learning based on graph regularization [2], collective classification of networked data [18], active learning for networked data [3] [19] and subspace learning in networks [14]. However, feature selection for networked data is rarely touched. Traditional feature selection methods are not suited for networked data because the i.i.d. assumption no longer holds.

In this paper, based on the above motivation, we present a supervised feature selection method for networked data. It is built upon Laplacian Regularized Least Squares (LapRLS) [2]. The key idea is to use the least squares regression to fit the labels with respect to the content information, and adopt graph regularization [20] [25] to utilize the link information. We study graph regularization on both undirected graph and directed graph. The basic assumption of graph regularization is that if two nodes are linked in a network (or one node links another), then their labels are likely to be the same. The proposed feature selection method aims at selecting a subset of features such that the empirical error of LapRLS is minimized. The resultant optimization problem is a mixed integer programming [4], which is difficult to solve. It is relaxed into a $L_{2,1}$-norm constrained LapRLS problem and solved by accelerated proximal gradient descent [16]. It is worth noting that we do not need to specify the number of features to select for the proposed feature selection method. It is implicitly controlled by a regularization parameter. Experiments on benchmark data sets indicate that the proposed method outperforms traditional feature selection method and the state-of-the-art learning-in-network approaches.

The remainder of this paper is organized as follows. In Section 2, we review traditional feature selection methods and learning-in-network approaches. We present the feature selection method in Section 3. The experiments on benchmark data sets are demonstrated in Section 4. Finally, we draw conclusions and point out some future work in Section 5.

1.1 Notation

The generic problem of supervised feature selection in network is as follows. Given a networked data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector of the i-th node, and $y_i \in \{1, 2, \ldots, c\}$ is the label of the i-th node, $\mathbf{A}$ is the adjacency matrix of the networked data, such that $A_{ij} = 1$ if there is a link from i-th node to the j-th node, and $A_{ij} = 0$ otherwise. For undirected graph, we have $A_{ij} = A_{ji}$. In other words, the adjacency matrix $\mathbf{A}$ is symmetric. For directed graph, the adjacency matrix $\mathbf{A}$ is asymmetric. The goal of feature selection is to find a feature subset of size m which contains the most informative features. We use $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ to represent the data matrix, and $\mathbf{Y} \in \mathbb{R}^{n\times c}$ to represent the target label matrix, where $Y_{ik} = 1$ if $y_i = k$, and $Y_{ik} = 0$ otherwise. Given a matrix $\mathbf{W} \in \mathbb{R}^{d\times m}$, we denote the i-th row of $\mathbf{W}$ by $\mathbf{w}^i$, and the j-th column of $\mathbf{W}$ by $\mathbf{w}_j$. The Frobenius norm of $\mathbf{W}$ is defined as $||\mathbf{W}||_F = \sqrt{\sum_i^d ||\mathbf{w}^i||_2^2}$, the $L_{2,0}$-norm of $\mathbf{W}$ is defined as $||\mathbf{W}||_{2,0} = \text{card}(||\mathbf{w}^1||_2, \ldots, ||\mathbf{w}^d||_2)$, and the $L_{2,1}$-norm of $\mathbf{W}$ is defined as $||\mathbf{W}||_{2,1} = \sum_i^d ||\mathbf{w}^i||_2$. $\mathbf{1}$ is a vector of all ones with an appropriate length. $\mathbf{0}$ is a vector of all zeros. $\mathbf{I}$ is an identity matrix with an appropriate size.

2. RELATED WORK

In this section, we give a brief review of traditional feature selection methods and existing learning methods in network respectively.

2.1 Feature Selection

In feature selection [23], the features may be scored either dependent or independent on a classifier. In general, there are three families of approaches to score them: filter-based, wrapper-based, and embedded methods [9]. Filter-based methods score the features as a pre-processing step, independently of the classifier. The most representative filter-based methods include Information Gain (IG), χ^2 [23], Fisher score [7] [8] and so on. Wrapper-based methods score the features according to their prediction performance when used with the classifier. Finally, embedded methods combine feature selection with the classifier. While the design of embedded methods is tightly coupled with the specific classifier, they are often considered as more effective than filters and wrappers [9]. It is worth noting that the proposed feature selection method for networked data belongs to the family of embedded method, because it is specifically designed for Laplacian Regularized Least Squares (LapRLS) [2]. Feature selection methods rely on search strategies to guide the search for the "best" feature subset. While a large number of search strategies can be used, one is often limited to the greedy (forward or backward) strategies. The search for the "best" features in the proposed method is in a principled way rather than greedy.

2.2 Learning in Network

Learning in network has received increasing interest in the past decade. [6] proposed a missing-link model, which generalizes probabilistic latent semantic analysis (PLSA) [11] to consider both content and link information. [26] proposed link-content matrix factorization (LCMF) method, which integrates content and link information into a matrix factorization framework. [13] proposed relation regularized matrix factorization (RRMF), which overcomes some limitations of LCMF. However, all the methods mentioned above are transductive learning methods [21]. Transductive learning methods work on the training and testing set together. They do not output a parametric classifier, hence they cannot generalize to new unseen data. When new data come, they need to re-learn based on the training and the testing data. This motivates inductive learning, which induces a parametric decision function in the whole sample space. [2] proposed Laplacian Regularized Least Squares (LapRLS) based on graph regularization. Although it is originally proposed for semi-supervised learning, it can be adapted to learning in network and works very well. [18] studied collective classification of networked data. [14] proposed prob-

abilistic relation principle component analysis (PRPCA), which is the sate-of-the-art subspace learning method in network. More recently, [3] [19] suggested active learning for networked data. On the other hand, there are some works such as modularity of network [17], which study how to measure the partition of a network. However, as far as we know, feature selection for networked data is still a rarely touched topic. This motivates the method presented in this paper.

3. THE PROPOSED METHOD

In this section, we will present a feature selection method for networked data. We first introduce Graph regularization and Laplacian Regularized Least Squares (LapRLS). Then we present the feature selection method in network, followed with its optimization algorithm and theoretical analysis.

3.1 Graph Regularization

Link information in network characterizes the structure of the network and relation between the nodes. In order to consider the link information, we turn to use *Graph Regularization* [20] [25], which is based on spectral graph theory [5]. Graph regularization has achieved great success in dimensionality reduction [1] [10] [22] and semi-supervised learning [27] [24] [2]. In the following, we will introduce graph regularization for both undirected graph and directed graph respectively.

3.1.1 Undirected Graph

For an undirected network, the basic assumption of undirected graph regularization is: if two nodes are linked together, then their labels are likely to be the same. It can be mathematically formulated as [20]

$$\begin{aligned}
&\frac{1}{2}\sum_{k=1}^{c}\sum_{ij}(\mathbf{w}_k^T\mathbf{x}_i-\mathbf{w}_k^T\mathbf{x}_j)^2A_{ij}\\
=&\sum_{k=1}^{c}\sum_{i,j}\mathbf{w}_k^T\mathbf{x}_iA_{ij}\mathbf{x}_i^T\mathbf{w}_k-\sum_{k=1}^{c}\sum_{i,j}\mathbf{w}_k^T\mathbf{x}_iA_{ij}\mathbf{x}_j^T\mathbf{w}_k\\
=&\sum_{k=1}^{c}\sum_{i}\mathbf{w}_k^T\mathbf{x}_iD_{ii}\mathbf{x}_i^T\mathbf{w}_k-\sum_{k=1}^{c}\sum_{i,j}\mathbf{w}_k^T\mathbf{x}_iA_{ij}\mathbf{x}_j^T\mathbf{w}_k\\
=&\sum_{k=1}^{c}\mathbf{w}_k^T\mathbf{X}(\mathbf{D}-\mathbf{W})\mathbf{X}^T\mathbf{w}_k\\
=&\mathrm{tr}(\mathbf{W}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{W}),
\end{aligned}\tag{1}$$

where $\mathbf{L}=\mathbf{D}-\mathbf{A}$ is the graph Laplacian [5] and $\mathbf{D}$ is a diagonal matrix called degree matrix with $D_{ii}=\sum_j A_{ij}$. Note that in undirected graph regularization, $\mathbf{A}$ is symmetric.

3.1.2 Directed Graph

For a directed graph, the undirected graph regularization in Eq.(1) does not work anymore since the adjacency matrix $\mathbf{A}$ is asymmetric. To revolve this problem, one way is to simply use $\mathbf{A}=\max(\mathbf{A},\mathbf{A}^T)$. Another way is to use directed graph regularization [25] instead. An edge of a directed graph is an ordered pair (i,j) where i and j are the node indices. The in-degree of the i-th node is defined as $D_i^-=\sum_{j\to i}A_{ji}$, where $j\to i$ denotes the j-th node has a directed link pointing to the i-th node, while out-degree of the i-th node is defined as $D_i^+=\sum_{i\to j}W_{ij}$, where $i\to j$ denotes the i-th node has a directed link pointing to the j-th node. Given the adjacency matrix $\mathbf{A}$ of a directed graph, we define a transition probability of random walk as $P_{ij}=A_{ij}/D_i^+$. It is obvious that it satisfies $\sum_j P_{ij}=1$. Assume the stationary distribution for i-th node is π_i. Then it satisfies $\sum_i\pi_i=1$ and $\pi_j=\sum_{i\to j}\pi_iP_{ij}$. The basic assumption of directed graph regularization is: if the i-th node links the j-th node, then their labels are likely to be the same. It is mathematically formulated as [25],

$$\begin{aligned}
&\frac{1}{2}\sum_{k=1}^{c}\sum_{(i,j)}(\mathbf{w}_k^T\mathbf{x}_i-\mathbf{w}_k^T\mathbf{x}_j)^2\pi_iP_{ij}\\
=&\frac{1}{4}\sum_{k=1}^{c}\sum_{j}(\sum_{i\to j}(\mathbf{w}_k^T\mathbf{x}_i-\mathbf{w}_k^T\mathbf{x}_j)^2\pi_iP_{ij}\\
+&\sum_{j\to i}(\mathbf{w}_k^T\mathbf{x}_j-\mathbf{w}_k^T\mathbf{x}_i)^2\pi_jP_{ji})\\
=&\sum_{k=1}^{c}\sum_{j}\mathbf{w}_k^T\mathbf{x}_j\pi_j\mathbf{x}_j^T\mathbf{w}_k-\frac{1}{2}\sum_{k=1}^{c}\sum_{j}(\sum_{i\to j}\mathbf{w}_k^T\mathbf{x}_i\pi_iP_{ij}\mathbf{x}_j\mathbf{w}_k^T\\
+&\sum_{j\to i}\mathbf{w}_k^T\mathbf{x}_j\pi_jP_{ji}\mathbf{x}_i\mathbf{w}_k^T)\\
=&\mathrm{tr}(\mathbf{W}^T\mathbf{X}(\mathbf{\Pi}-\frac{1}{2}(\mathbf{\Pi}\mathbf{P}+\mathbf{P}^T\mathbf{\Pi}))\mathbf{X}^T\mathbf{W})\\
=&\mathrm{tr}(\mathbf{W}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{W}),
\end{aligned}\tag{2}$$

where $\mathbf{L}=\mathbf{\Pi}-\frac{1}{2}(\mathbf{\Pi}\mathbf{P}+\mathbf{P}^T\mathbf{\Pi})$ is graph Laplacian of directed graph, $\mathbf{\Pi}$ is a diagonal matrix with $\Pi_{ii}=\pi_i$ and $\mathbf{P}$ is the transition matrix of random walk.

3.2 Laplacian Regularized Least Squares

Till now, we have introduced the graph regularization on both undirected and directed graph. Since the proposed feature selection method is built on Laplacian Regularized Least Squares (LapRLS), we will give a preliminary introduction here. In the setting of multi-class classification, LapRLS aims to learn c linear classifiers $f_k(\mathbf{x})=\mathbf{w}_k^T\mathbf{x}, 1\le k\le c$ by the following optimization problem,

$$\begin{aligned}
\arg\min_{\mathbf{W}}\quad&\frac{1}{2}||\mathbf{X}^T\mathbf{W}-\mathbf{Y}||_F^2+\frac{\lambda_A}{2}||\mathbf{W}||_F^2\\
+&\frac{\lambda_I}{2}\mathrm{tr}(\mathbf{W}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{W}),
\end{aligned}\tag{3}$$

where $\lambda_A,\lambda_I>0$ are positive regularization parameters, $\mathbf{W}=[\mathbf{w}_1,\ldots,\mathbf{w}_c]$. More specifically, the first term in the above objective function is traditional least squares, which is able to fit the input content information and labels. The second term is Frobenious norm regularization on $\mathbf{W}$ which controls the complexity of the linear classifier. The third term is graph regularization as we introduced before. It can be either undirected graph regularization in Eq. (1) or directed graph regularization in Eq.(2). It is used to encode the link information of the network (undirected or directed).

The above problem has a closed form solution

$$\mathbf{W}^*=(\mathbf{X}\mathbf{X}^T+\lambda_A\mathbf{I}+\lambda_I\mathbf{X}\mathbf{L}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{Y}.\tag{4}$$

Note that the inversion of the big matrix $\mathbf{X}\mathbf{X}^T+\lambda_A\mathbf{I}+\lambda_I\mathbf{X}\mathbf{L}\mathbf{X}^T$ is time consuming. Fortunately, it can be solved efficiently as a linear system equation.

Then the label of each data point $\mathbf{x}$ can be predicted by $l=\arg\max_k f_k(\mathbf{x})$.

3.3 Feature Selection in Network

In order to do feature selection, we introduce an indicator variable $\mathbf{p}$, where $\mathbf{p} = (p_1, \ldots, p_d)^T$ and $p_i \in \{0, 1\}, i = 1, \ldots, d$, to represent whether a feature is selected. We further introduce a diagonal matrix $\text{diag}(\mathbf{p})$. Then the input data matrix is now represented as $\text{diag}(\mathbf{p})\mathbf{X}$. In order to indicate that m features are selected, we constrain $\mathbf{p}^T\mathbf{1} = m$.

The proposed feature selection method aims at selecting a subset of features such that the empirical error of LapRLS is minimized. It can be mathematically formulated as

$$\begin{aligned} \arg\min_{\mathbf{p},\mathbf{W}} \quad & \frac{1}{2}||\mathbf{X}^T\text{diag}(\mathbf{p})\mathbf{W} - \mathbf{Y}||_F^2 + \frac{\lambda_A}{2}||\mathbf{W}||_F^2 \\ + \quad & \frac{\lambda_I}{2}\text{tr}(\mathbf{W}^T\text{diag}(\mathbf{p})\mathbf{X}\mathbf{L}\mathbf{X}^T\text{diag}(\mathbf{p})\mathbf{W}), \\ \text{s.t.} \quad & \mathbf{p} \in \{0,1\}^d, \mathbf{p}^T\mathbf{1} = m, \end{aligned} \tag{5}$$

where $\lambda_A, \lambda_I > 0$ are positive regularization parameters. It is worth noting that, when $\mathbf{p} = \mathbf{1}$, the proposed feature selection reduces to LapRLS in the input space. As a result, LapRLS can be seen as a special case of the proposed method without feature selection. We call Eq. (5) as *Feature Selection in Network* (FSNet). It is worth noting that the proposed method is heavily built upon LapRLS, so it is an embedded method.

As can be seen, the problem in Eq. (5) is mixed integer programming [4], which is difficult to solve. In the following, we will relax it into a continuous optimization problem and present an efficient algorithm.

Suppose we find the optimal solution of Eq. (5), i.e., $\mathbf{W}^*$ and $\mathbf{p}^*$, then $\mathbf{p}^*$ is a binary vector, and $\text{diag}(\mathbf{p})\mathbf{W}$ is a matrix where the elements of many rows are all zeros. This motivates us to absorb the indicator variables $\mathbf{p}$ into $\mathbf{W}$, and use $L_{2,0}$-norm on $\mathbf{W}$ to achieve feature selection, leading to the following problem

$$\begin{aligned} \arg\min_{\mathbf{W}} \quad & \frac{1}{2}||\mathbf{X}^T\mathbf{W} - \mathbf{Y}||_F^2 + \frac{\lambda_A}{2}||\mathbf{W}||_F^2 \\ + \quad & \frac{\lambda_I}{2}\text{tr}(\mathbf{W}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{W}), \\ \text{s.t.} \quad & ||\mathbf{W}||_{2,0} \leq m. \end{aligned} \tag{6}$$

Or equivalently the regularized problem,

$$\begin{aligned} \arg\min_{\mathbf{W}} \quad & \frac{1}{2}||\mathbf{X}^T\mathbf{W} - \mathbf{Y}||_F^2 + \frac{\lambda_A}{2}||\mathbf{W}||_F^2 \\ + \quad & \frac{\lambda_I}{2}\text{tr}(\mathbf{W}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{W}) \\ + \quad & \lambda||\mathbf{W}||_{2,0}, \end{aligned} \tag{7}$$

where $\lambda > 0$ is a regularization parameter. Note that it is difficult to give an analytical relationship between m and λ. Fortunately, such a relationship is not crucial for our problem. The objective function in Eq. (7) is a non-smooth and nonconvex function. We relax $||\mathbf{W}||_{2,0}$ to its convex hull [4], and obtain the following convex problem,

$$\begin{aligned} \arg\min_{\mathbf{W}} \quad & \frac{1}{2}||\mathbf{X}^T\mathbf{W} - \mathbf{Y}||_F^2 + \frac{\lambda_A}{2}||\mathbf{W}||_F^2 \\ + \quad & \frac{\lambda_I}{2}\text{tr}(\mathbf{W}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{W}) \\ + \quad & \lambda||\mathbf{W}||_{2,1}. \end{aligned} \tag{8}$$

In the following, we will present an algorithm for solving Eq. (8).

3.4 Proximal Gradient Descent

The most natural approach for solving the problem in Eq. (7) is the sub-gradient descent method [4]. However, its convergence rate is very slow, i.e., $O(\frac{1}{\epsilon^2})$ [16].

Recently, proximal gradient descent has received increasing attention in the machine learning community [12] [15]. It achieves the optimal convergence rate, i.e., $O(\frac{1}{\epsilon})$ for the first-order method and is able to deal with large-scale non-smooth convex problems. It can be seen as an extension of gradient descent, where the objective function to minimize is the composite of a smooth part and a non-smooth part. As to our problem, let

$$\begin{aligned} f(\mathbf{W}) &= \frac{1}{2}||\mathbf{X}^T\mathbf{W} - \mathbf{Y}||_F^2 + \frac{\lambda_A}{2}||\mathbf{W}||_F^2 \\ &+ \frac{\lambda_I}{2}\text{tr}(\mathbf{W}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{W}) \\ F(\mathbf{W}) &= f(\mathbf{W}) + \lambda||\mathbf{W}||_{2,1}. \end{aligned} \tag{9}$$

It is easy to show that $f(\mathbf{W})$ is convex and differentiable, while $\lambda||\mathbf{W}||_{2,1}$ is non-smooth but convex.

In each iteration of the proximal algorithm, $F(\mathbf{W})$ is linearized around the current estimate $\mathbf{W}_t$, and the value of $\mathbf{W}$ is updated as the solution of the following proximal gradient descent problem,

$$\mathbf{W}_{t+1} = \arg\min_{\mathbf{W}} G_{\eta_t}(\mathbf{W}, \mathbf{W}_t), \tag{10}$$

where $G_{\eta_t}(\mathbf{W}, \mathbf{W}_t)$ is defined as

$$\begin{aligned} G_{\eta_t}(\mathbf{W}, \mathbf{W}_t) &= f(\mathbf{W}_t) + \langle \nabla f(\mathbf{W}_t), \mathbf{W} - \mathbf{W}_t \rangle \\ &+ \frac{\eta_t}{2}||\mathbf{W} - \mathbf{W}_t||^2 + \lambda||\mathbf{W}||_{2,1}. \end{aligned} \tag{11}$$

In our problem, we have

$$\nabla f(\mathbf{W}_t) = \mathbf{X}\mathbf{X}^T\mathbf{W}_t - \mathbf{X}\mathbf{Y} + \lambda_A\mathbf{W}_t + \lambda_I\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{W}_t. \tag{12}$$

The philosophy under this formulation is that if the optimization problem in Eq. (10) can be solved by exploiting the structure of the $L_{2,1}$ norm, then the convergence rate of the resulting algorithm is the same as that of gradient descent method, i.e., $O(\frac{1}{\epsilon})$, since no approximation on the non-smooth term is employed.

By ignoring the terms in $G_{\eta_t}(\mathbf{W}, \mathbf{W}_t)$ that is independent of $\mathbf{W}$, the optimization problem in Eq. (10) boils down to

$$\mathbf{W}_{t+1} = \pi_{\eta_t}(\mathbf{W}_t) = \arg\min_{\mathbf{W}} \frac{1}{2}||\mathbf{W} - \mathbf{U}_t||_F^2 + \frac{\lambda}{\eta_t}||\mathbf{W}||_{2,1}, \tag{13}$$

where $\mathbf{U}_t = \mathbf{W}_t - \frac{1}{\eta_t}\nabla f(\mathbf{W}_t)$. It can be further decomposed into c separate subproblems of dimension d

$$\mathbf{w}_{t+1}^i = \arg\min ||\mathbf{w}^i - \mathbf{u}_t^i||_2^2 + \frac{\lambda}{\eta_t}||\mathbf{w}^i||_2, \tag{14}$$

where $\mathbf{w}_{t+1}^i$, $\mathbf{w}^i$ and $\mathbf{u}_t^i$ are the i-th rows of $\mathbf{W}_{t+1}$, $\mathbf{W}$ and $\mathbf{U}_t$ respectively. It has a closed form solution [15] as follows

$$\mathbf{w}^{i*} = \begin{cases} (1 - \frac{\lambda}{\eta_t||\mathbf{u}_t^i||})\mathbf{u}_t^i, & \text{if } ||\mathbf{u}_t^i|| > \frac{\lambda}{\eta_t} \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{15}$$

Thus, the proximal gradient decent in Eq. (10) has the same convergence rate of $O(\frac{1}{\epsilon})$ as gradient descent for smooth problem.

3.5 Accelerated Proximal Gradient Decent

To achieve more efficient optimization, we employ Nesterov's method [16] to accelerate the proximal gradient decent in Eq. (10), which owns the convergence rate as $O(\frac{1}{\sqrt{\epsilon}})$. More specifically, we construct a linear combination of $\mathbf{W}_t$ and $\mathbf{W}_{t+1}$ to update $\mathbf{V}_{t+1}$ as follows:

$$\mathbf{V}_{t+1} = \mathbf{W}_t + \frac{\alpha_t - 1}{\alpha_{t+1}}(\mathbf{W}_{t+1} - \mathbf{W}_t), \tag{16}$$

where the sequence $\{\alpha_t\}_{t\geq 1}$ is conventionally set to be $\alpha_{t+1} = \frac{1+\sqrt{1+4\alpha_t^2}}{2}$. For more detail, please refer to [12]. Here we directly present the final algorithm for optimizing Eq. (8) in Algorithm 1.

Algorithm 1 Feature Selection in Network

Initialize: $\eta_0, \mathbf{W}_1 \in \mathbb{R}^{d\times m}, \alpha_1 = 1$;
repeat
  while $F(\pi_{\eta_{t-1}}(\mathbf{W}_t)) > G_{\eta_{t-1}}(\pi_{\eta_{t-1}}(\mathbf{W}_t), \mathbf{W}_t)$ **do**
    Set $\eta_{t-1} = \gamma\eta_{t-1}$
  end while
  Set $\eta_t = \eta_{t-1}$
  Compute $\mathbf{W}_{t+1} = \arg\min_{\mathbf{W}} G_{\eta_t}(\mathbf{W}, \mathbf{V}_t)$
  Compute $\alpha_{t+1} = \frac{1+\sqrt{1+4\alpha_t^2}}{2}$
  Compute $\mathbf{V}_{t+1} = \mathbf{W}_t + \frac{\alpha_t - 1}{\alpha_{t+1}}(\mathbf{W}_{t+1} - \mathbf{W}_t)$
until convergence

3.6 Discussion

Once we obtain $\mathbf{W}$, we can obtain the selected features as follows. We calculate the score for each feature as

$$\text{score}(i) = \sqrt{\sum_j W_{i,j}^2}. \tag{17}$$

Then we select all those features whose score is nonzero.

Given the selected features, we can train a classifier such as regularized least squares (RLS) [7] or LapRLS on the reduced data to do classification. Since the proposed method is specifically designed for LapRLS, we use LapRLS as the classifier in our experiments. More importantly, we do not need to re-train a LapRLS using the selected features. We can use the learnt weight matrix $\mathbf{W}$ directly, because it is already a linear classifier based on the selected features.

It can be seen that one additional advantage of the proposed method is that we do not need to predefine the number of selected features, i.e., m. In fact, the number of features to select is implicitly controlled by the regularization parameter λ.

3.7 Convergence Analysis

The convergence property of Algorithm 1 is stated in the following theorem.

Theorem 3.1 *[16] Let $\{\mathbf{W}_t\}$ be the sequence generated by Algorithm 1, then for any $t \geq 1$ we have*

$$F(\mathbf{W}_t) - F(\mathbf{W}^*) \leq \frac{2\gamma L||\mathbf{W}_1 - \mathbf{W}^*||_F^2}{(t+1)^2}, \tag{18}$$

where L is the Lipschitz constant of the gradient of $f(\mathbf{W})$ in the objective function, $\mathbf{W}^* = \arg\min_{\mathbf{W}} F(\mathbf{W})$.

Theorem 3.1 shows that the convergence rate of the accelerated proximal gradient method is $O(\frac{1}{\sqrt{\epsilon}})$. The detail proof of the above theorem can be found in [16] [12].

4. EXPERIMENTS

In this section, we will evaluate the proposed method for web page classification and research paper classification. The task of the experiments is to classify the networked data based on their content information and link structure. We compare it with some state-of-the-art methods.

4.1 Data Sets

We use the same data sets[1] used in [26] to evaluate our method.

WebKB data set contains about 6,000 web pages collected from the web sites of computer science departments of four universities (Cornell, Texas, Washington, and Wisconsin). Each web page is labeled with one out of seven categories: student, professor, course, project, staff, department, and "other". The characteristics about the WebKB data set are briefly summarized in Table 1.

Table 1: Description of the WebKB data set

Data Sets	#samples	#features	#links	#classes
Cornell	827	4134	1626	7
Texas	814	4029	1480	7
Washington	1166	4165	2218	7
Wisconsin	1210	4189	3200	6

Cora data set contains the abstracts and references of about 34,000 research papers from the computer science community. The task is to classify each paper into one of the subfields of data structure (DS), hardware and architecture (HA), machine learning (ML), and programming language (PL). The characteristics about the Cora data set are summarized in Table 2.

Table 2: Description of the Cora data set

Data Sets	#samples	#features	#links	#classes
DS	751	6234	1283	9
HA	400	3989	793	7
ML	1617	8329	4046	7
PL	1575	7949	4918	9

We can see that the dimensionality of the data is very high (thousands of words). In addition, the link in the above two data sets is in nature directed. In the following experiments, we study both undirected graph regularization and directed graph regularization. To use undirected graph regularization, we use a symmetric adjacency matrix, i.e., $\mathbf{A} = \max(\mathbf{A}, \mathbf{A}^T)$.

4.2 Methods

In order to evaluate the proposed feature selection methods, we compare it with the following baselines.

The first baseline is regularized least squares (RLS) [7], which is the state-of-the-art classifier for i.i.d. data and is very related to LapRLS. In detail, RLS is applied in the following 3 settings.

[1] http://www.nec-labs.com/~zsh/files/link-fact-data.zip

- **RLS on content**: It ignores the link structure in the data, and applies RLS only on the content information in the original bag-of-words representation.
- **RLS on link**: It ignores the content information, and treats links as the features, i.e, the i-th feature is link to the i-th page.
- **RLS on content+link**: The content features and link features of the two methods above are concatenated to give the feature representation, then RLS is applied.

The second baseline is LapRLS [2], which is able to utilize both content information and link information. It can be seen as a special case of the proposed method. That is, it does not do feature selection. So it uses all the original features. In detail, we adopt LapRLS in the following 2 scenarios.

- **undirected LapRLS**: It is LapRLS with undirected graph regularization in Eq.(1). We treat the networked data as undirected graph. That is, we set $\mathbf{A} = \max(\mathbf{A}, \mathbf{A}^T)$.
- **directed LapRLS**: It is LapRLS with directed graph regularization in Eq.(2).

The third baseline is a traditional feature selection method. We choose Fisher score [7] because a previous study [8] showed that it is comparable to or even better than the other feature selection methods [23]. We first apply Fisher score to select features, then apply RLS in 3 ways as before. As a result, we get three methods: **FS+RLS on content**, **FS+RLS on link**, and **FS+RLS on content+link**. In addition, we are also interested in first applying Fisher score on the content information to select features, and then apply **undirected LapRLS** and **directed LapRLS**. We name these two methods as **FS+undirected LapRLS** and **FS+directed LapRLS**

The fourth baseline is probabilistic relation principle component analysis (PRPCA) [14] which is proposed recently. It is the state-of-the-art subspace learning method in networks. As we know, both subspace learning and feature selection can achieve dimensionality reduction. As a result, we are very interested in comparing the proposed method with PRPCA.

For the proposed feature selection method, i.e., FSNet, similar with LapRLS, we also evaluate it in two settings: **undirected FSNet** and **directed FSNet**. Note that we do not need to train a classifier after selecting the features, because FSNet not only selects the features, but also outputs a linear classifier, i.e., $\mathbf{W}$.

We did not compare the proposed method with the methods proposed in [25] [26] and [13] because these two methods are transductive learning methods [21] rather than inductive learning methods. In other words, those methods utilize the testing data when training. Hence it is unfair to do comparison.

4.3 Parameter Settings

For RLS, we set $\lambda_A = 1$ on WebKB data set and $\lambda_A = 0.1$ on Cora Data set.

For undirected LapRLS and directed LapRLS, we simply fix $\lambda_A = 1$, and tune λ_I by searching the grid $\{0.001, 0.005, \ldots, 0.05, 0.1\}$.

For Fisher score, we select $\{50\%, 60\%, \ldots, 90\%\}$ of the original features (content, link, or content+link), and the best result is reported.

For PRPCA, according to [14], we set the dimensionality of the subspace to 50 and the number of iterations to 5.

For undirected FSNet and directed FSNet, we fix $\lambda_A = 1$, and tune λ_I by searching the grid $\{0.001, 0.005, \ldots, 0.05, 0.1\}$. In addition, on the WebKB data set, we set $\lambda = 1$, while on the Cora data set, we use $\lambda = 0.001$. In the accelerated proximal gradient descent, there is a parameter γ. We simply set $\gamma = 1.1$.

We randomly split data into five folds and repeat the experiment for five times. For each time, we use one fold for testing, and the other four for training.

4.4 Study on Convergence

Before reporting the classification results, in this subsection, we first examine the convergence of the accelerated proximal gradient descent in Algorithm 1 and original proximal gradient descent. In Figure 2, we plot the objective function value in Eq. (8) with respect to the number of iterations on the WebKB (Cornell) data subset. The parameter setting is $\lambda_A = 1, \lambda = 1, \lambda_I = 0.01$. In the figure, the y-axis is the value of objective function and the x-axis denotes the iteration number.

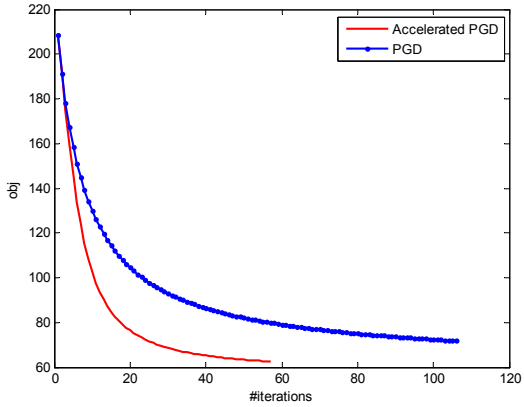


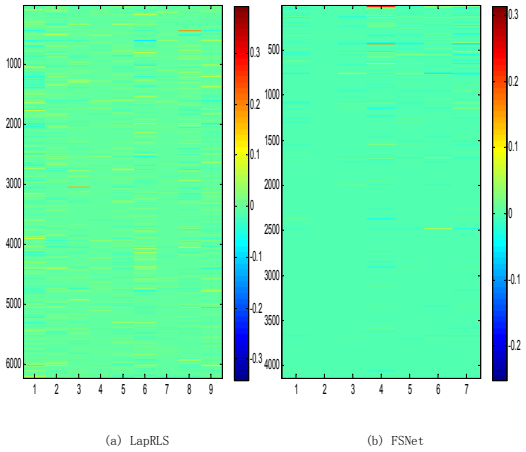
Figure 2: The objective function value of FSNet with respect to the number of iterations for (a) accelerated proximal gradient descent and (b) proximal gradient descent on the WebKB (Cornell) data subset.

We can see that the accelerated proximal gradient descent (Accelerated PGD) converges faster than proximal gradient descent (PGD). In detail, the Accelerated PGD usually converges within 50 iterations, while the original PGD needs more than 100 iterations to converge on WebKB data sets. This is consistent with the theoretical result in Section 3.7. That is, the convergence rate of Accelerated PGD is $O(\frac{1}{\sqrt{\epsilon}})$, which is faster than that of PGD, i.e., $O(\frac{1}{\epsilon})$. Similar results can be observed on other data sets.

The experimental results of FSNet in the rest part are all achieved by accelerated proximal gradient descent. Note that given a sufficient number of iterations, both accelerated proximal gradient descent and proximal gradient descent will converge to the same solution, because the optimization problem of FSNet in Eq. (8) is convex.

4.5 Study on the Weight Matrix W

To get a better understanding of our approach, we plot the learnt weight matrix, i.e., **W** of our method and LapRLS on WebKB (Cornell) data subset in Figure 3. In the figure, (a) shows the weight matrix of LapRLS, while (b) shows the weight matrix of FSNet. The parameter settings are $\lambda_A = 1, \lambda_I = 0.01$ for LapRLS, and $\lambda_A = 1, \lambda_I = 0.01, \lambda = 1$ for FSNet. It can be seen that many rows of the weight matrix of FSNet are all zero, which leads to feature selection. We call it "row-sparsity". That means, the corresponding features are eliminated. In contrast, the weight matrix of LapRLS is not row-sparse, which is not able to do feature selection.



(a) LapRLS (b) FSNet

Figure 3: The weight matrix W of the learnt classifier: (a) LapRLS; (b) FSNet on the WebKB (Cornell) data subset. For better viewing, please see it in colored pdf.

4.6 Classification Results

The classification results on the WebKB data set and the Cora data set are reported in Tables 3 and 4 respectively. For clear comparison, we also show the results in Figures 4 and 5.

It can be observed that:

1. The proposed method outperforms the baseline methods consistently on the WebKB data set, while the proposed method outperforms the baseline methods on 3 out of 4 subsets on the Cora data set.

2. The proposed method outperforms LapRLS, which can be seen as a special case of the proposed method without feature selection. This indicates that feature selection is crucial and beneficial for classification of networked data.

3. FS+LapRLS is inferior to FSNet. This is because that Fisher score is based on i.i.d. assumption, while the networked data violate this assumption. In contrast, FSNet is able to utilize both content and link information in a principled way. On the other hand, FS+LapRLS is no better than LapRLS. This is because Fisher score in FS+LapRLS cannot consider the link information, which is crucial for learning in network.

4. PRPCA also achieves very good result on WebKB data set. However, it is not as good as FSNet. The reason is that PRPCA is an unsupervised subspace learning method, while FSNet is a supervised method. Supervised dimensionality reduction methods are generally better than unsupervised methods for classification.

5. The reason why the proposed method is not as good as some baseline methods (RLS on content+link) on the Cora data set may be that the citation relation among papers is more complicated than hyper-link among web pages. Graph regularization may not be that useful to utilize the underlying information of citation. We will investigate other techniques to fully use the citation information in the future.

6. The proposed methods with undirected graph regularization and directed graph regularization achieve comparable results. We deem that FSNet with directed graph regularization will achieve better result than FSNet with undirected graph regularization because the link information in the two data sets is directed. However, they eventually achieve very similar results. As far as we know, there is no theoretical work on the comparison between undirected and directed graph regularization. This needs further study.

4.7 Sensitivity vs. the Regularization Parameter λ_I

FSNet has three parameters, which are the regularization parameters, i.e., λ_A, λ_I and λ. In our experiments, we simply set $\lambda_A = 1$, and $\lambda = 1$ on WebKB data set while $\lambda = 0.001$ on Cora Data set. The only parameter needed to tune is λ_I. It controls the contribution of link information. Hence we investigate the classification accuracy with respect to the regularization parameter λ_I. We vary the value of λ_I, and plot the corresponding classification accuracy on the WebKB and Cora data sets in Figures 6 and 7 respectively.

As can be seen, FSNet is not sensitive to the regularization parameter λ_I in a wide range of λ_I. In detail, FSNet achieves consistently better performance than LapRLS with the λ_I varying from 0.001 to 0.1 on both data sets. It is a very appealing property because we do not need to tune the regularization parameter painfully in the application.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a feature selection method based on Laplacian Regularized Least Squares (LapRLS) for the data distributed in a network. We use linear regression to fit the content information, and use graph regularization to consider the link information. Then the feature selection method is casted as selecting a subset of features such that the training error of LapRLS is minimized. The resultant optimization problem is a mixed integer programming, which is difficult to solve. It is relaxed into a $L_{2,1}$-norm constrained least squares problem and solved by accelerated proximal gradient algorithm. Experiments on the benchmark networked data show that the proposed method outperforms the state-of-the-art methods.

For the future work, we will study other kinds of techniques to utilize link information. For example, [17] proposed modularity to measure the strength of a partition for real-world networks by taking into account the degree distribution of nodes. It has been shown effective in various kinds of complex networks. We will investigate how to incorporate modularity into our method.

Table 3: Classification Accuracy (%) on the four subsets of WebKB data set

Data Sets	Cornell	Texas	Washington	Wisconsin
RLS on content	84.12±2.16	78.15±3.23	86.78±2.22	84.30±2.19
RLS on links	28.48±3.71	28.52±4.46	23.52±1.34	34.88±2.73
RLS on content+link	84.36±1.45	80.00±2.60	87.98±1.82	85.29±1.79
FS+RLS on content	84.12±2.07	78.27±2.97	86.09±2.03	84.05±2.18
FS+RLS on link	27.39±3.62	26.79±5.88	23.18±1.66	27.52±2.46
FS+RLS on content+link	84.85±1.42	79.75±3.64	87.73±2.24	85.12±2.13
undirected LapRLS	84.24±2.10	78.40±2.39	86.78±2.07	84.55±2.40
directed LapRLS	85.21±1.75	78.15±3.34	86.87±2.12	84.71±2.28
FS+undirected LapRLS	84.24±2.39	78.89±2.28	86.35±2.18	84.05±2.14
FS+directed LapRLS	85.21±1.95	78.40±3.18	86.18±1.95	84.13±2.27
PRPCA	86.30±2.89	81.85±1.83	89.87±1.62	88.93±2.03
undirected FSNet	**88.24±1.58**	84.69±1.34	**92.02±2.09**	**89.09±2.16**
directed FSNet	88.12±1.64	**84.81±1.61**	**92.02±2.09**	89.01±1.95

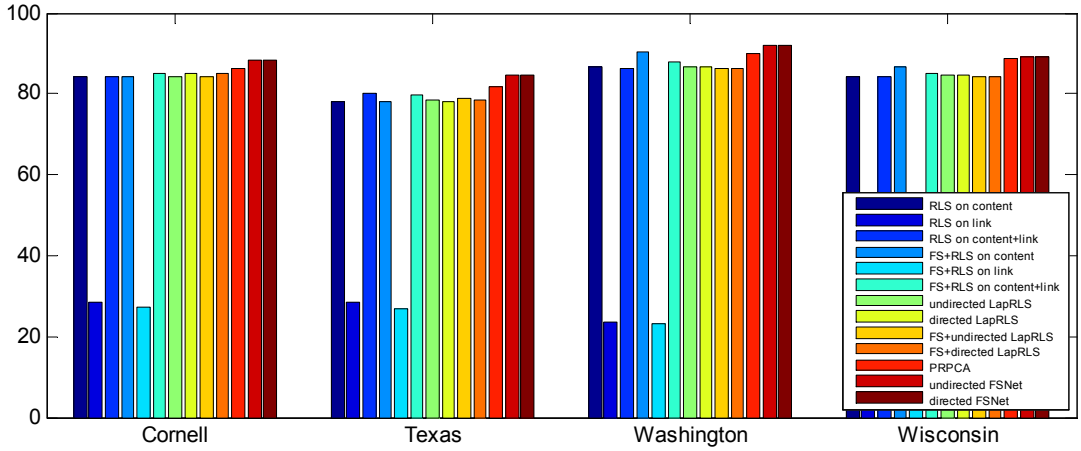


Figure 4: Classification accuracy on the four subsets of WebKB dataset.

Table 4: Classification Accuracy (%) on the four subsets of Cora data set

Data Sets	DS	HA	ML	PL
RLS on content	52.80±7.47	66.00±4.79	63.03±1.72	55.37±1.86
RLS on links	54.13±2.18	52.75±9.50	54.80±3.68	52.83±2.38
RLS on content+link	59.47±5.19	**79.50±5.05**	70.09±2.87	60.83±2.30
FS+RLS on content	54.00±5.60	66.00±4.79	63.10±1.30	55.37±1.86
FS+RLS on link	54.13±2.18	52.50±9.14	54.80±3.68	53.02±2.18
FS+RLS on content+link	59.73±6.51	75.50±7.10	70.09±2.87	60.83±2.30
undirected LapRLS	57.47±6.15	65.75±5.84	67.24±1.86	58.22±1.53
directed LapRLS	57.33±6.25	66.00±4.79	67.18±2.20	58.29±2.64
FS+undirected LapRLS	57.47±6.15	65.75±5.84	67.31±2.21	57.65±2.25
FS+directed LapRLS	57.33±6.25	66.25±4.59	67.37±1.99	58.10±1.88
PRPCA	53.07±4.80	68.25±3.60	68.36±2.41	54.98±2.41
undirected FSNet	**61.60±5.30**	67.50±5.59	**72.38±1.96**	61.52±2.92
directed FSNet	**61.60±5.30**	67.00±5.63	**72.38±1.96**	**61.52±2.59**

Acknowledgements

The work was supported in part by NSF IIS-09-05215, U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265, and the U.S. Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. We thank the anonymous reviewers for their helpful comments.

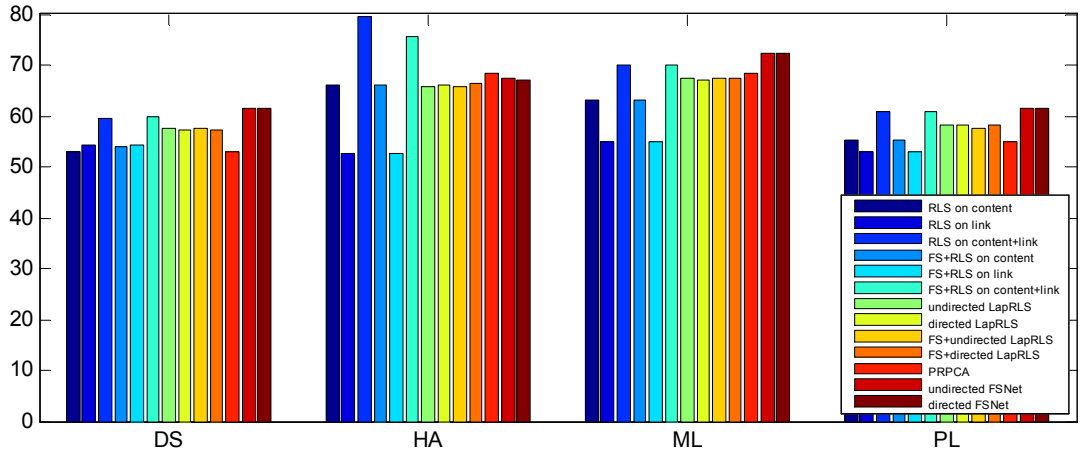
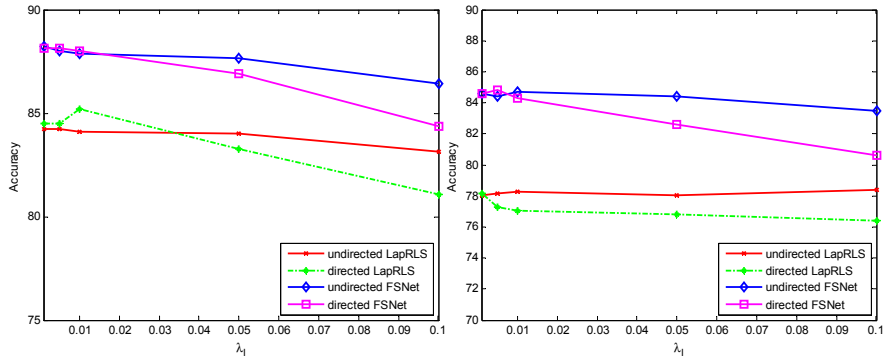
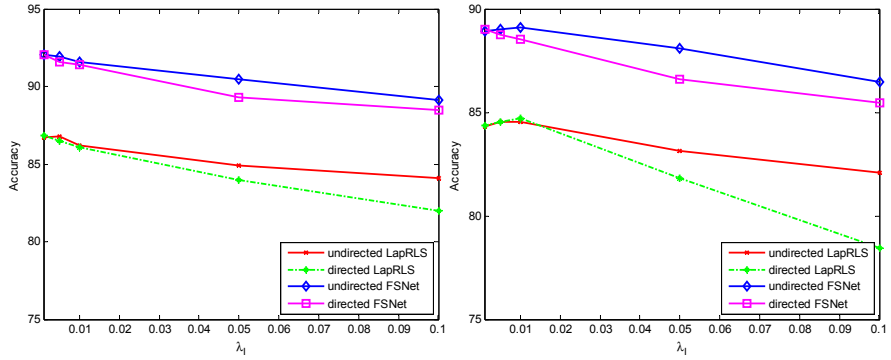


Figure 5: Classification accuracy on the four subsets of Cora dataset.



(a) Cornell (a) Texas



(c) Washington (d) Wisconsin

Figure 6: Classification accuracy of LapRLS and FSNet with respect to the regularization parameter λ_I on the four subsets of WebKB dataset.

6. REFERENCES

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[2] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[3] M. Bilgic, L. Mihalkova, and L. Getoor. Active learning for networked data. In *ICML*, pages 79–86, 2010.

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

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

[6] D. A. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *NIPS*, pages 430–436, 2000.

[7] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2001.

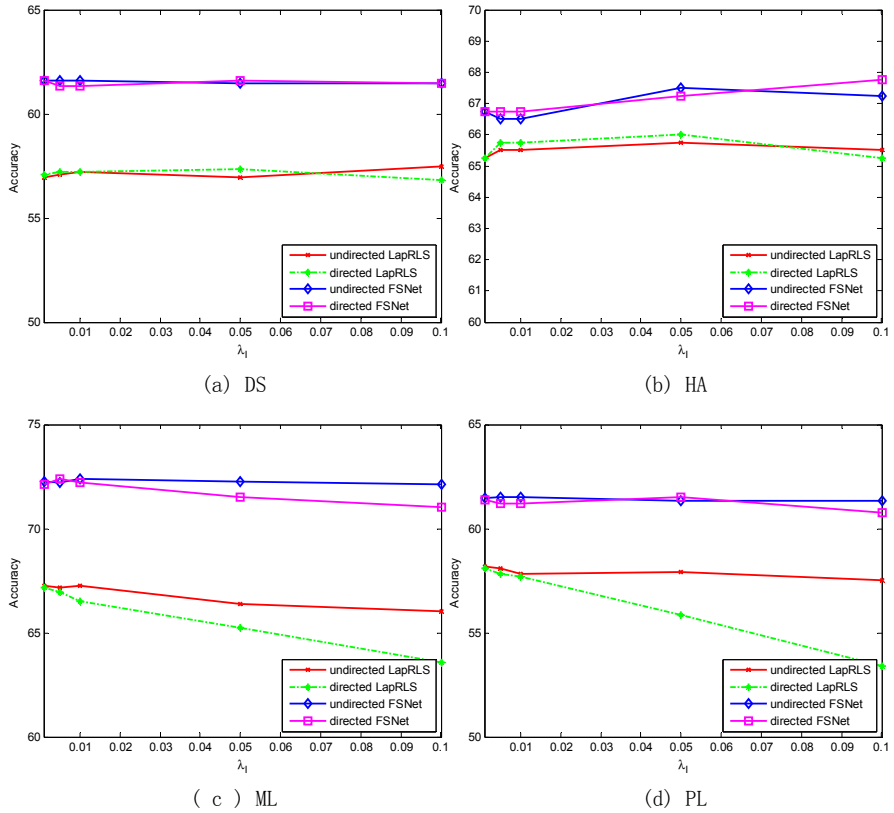


Figure 7: Classification accuracy of LapRLS and FSNet with respect to the regularization parameter λ_I on the four subsets of Cora dataset.

[8] Q. Gu, Z. Li, and J. Han. Generalized fisher score for feature selection. In *UAI*, 2011.

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

[10] X. He and P. Niyogi. Locality preserving projections. In *NIPS*, 2003.

[11] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196, 2001.

[12] S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *ICML*, page 58, 2009.

[13] W.-J. Li and D.-Y. Yeung. Relation regularized matrix factorization. In *IJCAI*, pages 1126–1131, 2009.

[14] W.-J. Li, D.-Y. Yeung, and Z. Zhang. Probabilistic relational pca. In *NIPS*.

[15] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient $l_{2,1}$-norm minimization. In *UAI*, 2009.

[16] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2003.

[17] M. E. Newman. Modularity and community structure in networks. *Proc Natl Acad Sci U S A*, 103(23):8577–8582, June 2006.

[18] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[19] L. Shi, Y. Zhao, and J. Tang. Combining link and content for collective active learning. In *CIKM*, pages 1829–1832, 2010.

[20] A. J. Smola and R. I. Kondor. Kernels and regularization on graphs. In *COLT*, pages 144–158, 2003.

[21] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.

[22] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):40–51, 2007.

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

[24] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2003.

[25] D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *ICML*, pages 1036–1043, 2005.

[26] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *SIGIR*, pages 487–494, 2007.

[27] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.