Unsupervised Feature Selection on Networks: A Generative View

Xiaokai Wei[*], **Bokai Cao**[*] **and Philip S. Yu**[*†]
[*]Department of Computer Science, University of Illinois at Chicago, IL, USA
[†]Institute for Data Science, Tsinghua University, Beijing, China
{xwei2,caobokai,psyu}@uic.edu

Abstract

In the past decade, social and information networks have become prevalent, and research on the network data has attracted much attention. Besides the link structure, network data are often equipped with the content information (*i.e*, node attributes) that is usually noisy and characterized by high dimensionality. As the curse of dimensionality could hamper the performance of many machine learning tasks on networks (*e.g.*, community detection and link prediction), feature selection can be a useful technique for alleviating such issue. In this paper, we investigate the problem of unsupervised feature selection on networks. Most existing feature selection methods fail to incorporate the linkage information, and the state-of-the-art approaches usually rely on pseudo labels generated from clustering. Such cluster labels may be far from accurate and can mislead the feature selection process. To address these issues, we propose a generative point of view for unsupervised features selection on networks that can seamlessly exploit the linkage and content information in a more effective manner. We assume that the link structures and node content are generated from a succinct set of high-quality features, and we find these features through maximizing the likelihood of the generation process. Experimental results on three real-world datasets show that our approach can select more discriminative features than state-of-the-art methods.

Introduction

Network data have become increasingly popular in the past decade, because of the proliferation of various social and information networks. Social networks such as Facebook and Twitter have millions of users all across the world. Different forms of information networks, *e.g.*, co-author networks, citation networks and protein interaction networks, also attract considerable research attention (Newman and Girvan 2004) (Backstrom and Leskovec 2011). In addition to the link structure, these network data are usually accompanied with content information on the nodes. For example, one can extract thousands of profiling features for users in social networks or ontology features for genes in protein interaction networks. However, redundant and irrelevant features might be included in the high-dimensional feature space. Feature selection (He, Cai, and Niyogi 2005) (Nie et al. 2010) is a useful technique since it can help alleviate the curse of dimensionality, speed up the learning process and provide better interpretability. However, not much research effort exists to explore feature selection on networks, especially in unsupervised scenario.

Depending on the availability of class labels, feature selection algorithms can be categorized into supervised methods and unsupervised methods. In the supervised setting, class labels provide a clear guidance to the feature selection process. In the unsupervised setting, feature selection becomes more challenging due to the lack of class labels. In this paper, we focus on unsupervised feature selection as class labels are usually expensive to obtain. State-of-the-art approaches introduce the notion of pseudo labels (Yang et al. 2011) (Li et al. 2012) (Qian and Zhai 2013) to guide the feature selection process. The basic idea is to imitate supervised methods by generating pseudo-labels via certain clustering methods (*e.g.*, spectral clustering and non-negative matrix factorization), and performing sparse regression towards these cluster labels. However, the generated pseudo labels are usually inaccurate and could further mislead the feature selection process.

Moreover, traditional feature selection approaches assume that data instances are independent and identically distributed (i.i.d). In the network data, however, instances are implicitly or explicitly related with certain correlations and dependencies. For example, in research collaboration networks, researchers who collaborate with each other (*i.e.*, connections in the network) tend to share more similar research topics (*i.e.*, close distances in the feature space) than researchers without such collaboration. Most existing feature selection approaches fail to exploit the rich information contained in the links.

Motivated by the importance of feature selection on networks and the deficiency of existing approaches, we propose a novel unsupervised feature selection method from a generative point of view. Our aim is to effectively incorporate information from both link structures and node attributes in the network data. Rather than using potentially inaccurate pseudo labels to guide the feature selection process, we assume that link structures and node attributes are generated by an oracle set of features. We propose a probabilistic model for this generative process. By performing inference using the linkage and attribute information, we can recover a succinct set of high-quality features. In this manner, we utilize information directly from the network data without generating intermediate pseudo labels. We refer to the proposed approach as Generative Feature Selection (GFS). To our knowledge, no existing method has adopted a generative view for feature selection.

As the state-of-the-art approaches on unsupervised feature selection are mostly pseudo label based methods, we illustrate the essential differences of these approaches and our approach in Figure 1. The class labels can be viewed as a perfect summarization of the data and using them to guide feature selection can usually achieve good performance (Figure 1a). Pseudo label based approaches attempt to first summarize the information from the data via clustering, and the pseudo labels serve as a proxy between the original data and the selected features (Figure 1b). However, such inaccurate summarization loses much information of the data. Our approach avoids the intermediate step and directly builds connections between the original data and the selected features (*i.e.*, oracle features). As a result, more information from the data could be utilized to guide the feature selection process (Figure 1c).

Related Work

Feature Selection for Traditional Data

Feature selection aims to select the most relevant ones from a large number of features and traditional feature selection methods generally fall into three categories: filter models (Zhao and Liu 2007) (Peng, Long, and Ding 2005), wrapper models (Dy and Brodley 2004) and embedded models (Cawley, Talbot, and Girolami 2006) (Tibshirani 1996).

Our work focuses on unsupervised scenario as class labels are usually expensive to obtain. One popular guiding principle for unsupervised feature selection is to preserve the local manifold structure or similarity (He, Cai, and Niyogi 2005) (Zhao and Liu 2007) (Zhao, Wang, and Liu 2010). Recently, pseudo label based frameworks (Yang et al. 2011) (Li et al. 2012) (Qian and Zhai 2013) have gained much popularity. Unsupervised Discriminative Feature Selection (UDFS) (Yang et al. 2011) introduces pseudo labels to better capture the discriminative information and the sparsity-inducing $L_{2,1}$ norm is used to select features in an iterative manner. NDFS (Li et al. 2012) performs non-negative spectral analysis and feature selection simultaneously. RUFS (Qian and Zhai 2013) and RSFS (Shi, Du, and Shen 2014) utilizes robust learning framework for generating pseudo labels. Essentially, different pseudo label based methods all use a $L_{2,1}$-regularized regression based framework with different clustering algorithms and constraints on pseudo labels. Since the clustering label is usually far from the groundtruth, it could result in degenerated quality of selected features.

Feature Selection for Network Data

In recent years, efforts have been made towards feature selection on network data. (Gu and Han 2011) (Tang and Liu 2012a) address supervised feature selection on network data via adding network-based regularization term to enforce similarity between neighbors. In unsupervised scenario, POPFS (Wei, Xie, and Yu 2015) uses network links to guide feature selection efficiently but it fails to use content information. Linked Unsupervised Feature Selection (LUFS) (Tang and Liu 2012b) is the only unsupervised feature selection method that utilizes both content and link information. LUFS exploits network information through incorporating social dimension based regularization (Tang and Liu 2009) into the UDFS framework (Yang et al. 2011). It enforces the nodes within the same social dimension to have similar pseudo labels. But the social dimensions generated from links (*e.g.*, by modularity (Newman 2006) or spectral clustering (Ng, Jordan, and Weiss 2001)) and pseudo labels generated from attributes are usually far from accurate, which could mislead the feature selection process.

Problem Formulation

Preliminaries

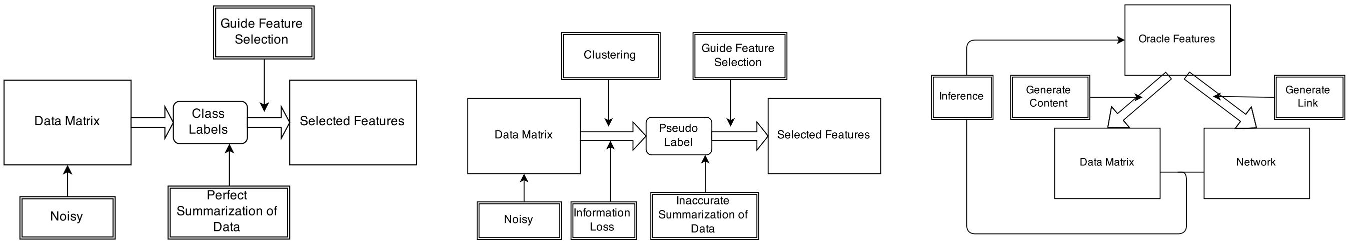
In this section, we present several concepts as preliminaries of our unsupervised feature selection method. In the rest of the paper, we use features and attributes interchangeably. Our goal is to select a set of important features on the network with node attributes, which we refer to as *attributed network*.

Definition 1 (Attributed Network) *An attributed network $G = (V, E, X)$ consists of V, the set of nodes, $E \subseteq V \times V$, the set of links, and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ where $n = |V|$ and $\mathbf{x}_i \in \mathbb{R}^D$ is the feature/attribute vector of the node v_i.*

In the supervised setting, one can select discriminative features that provide good separability of different classes. For unsupervised feature selection, there is no such clear guidance due to the lack of labels. Instead of relying on inaccurate pseudo labels, we aim to directly exploit the information from the data. From a generative point of view, we assume that link structures and node features are generated by an oracle set of features. Our goal is to recover this set of features through inference on the network. Specifically, we assume that there are $d \ll D$ important features among all features which are referred to as *oracle features*. All the node content and the network links are generated by these d oracle features. We use $\mathbf{s} = \{0, 1\}^D$ as the indicator vector for oracle features, where s_p equals 1 if the p-th feature is an oracle feature and 0 otherwise. Let us denote the diagonal matrix with diagonal elements $\mathbf{s}$ as $\text{diag}(\mathbf{s})$. Therefore, the oracle feature vector of the node v_i is $\text{diag}(\mathbf{s})\mathbf{x}_i$.

Modeling Link Generation

Most unsupervised feature selection methods cannot exploit linkage information. In our generative framework, we can incorporate linkage information seamlessly. From a generative point of view, we assume that the links are generated from a set of oracle features. More specifically, we assume that the probability of a link is determined by the oracle affinity between two nodes defined as follows.



(a) Supervised feature selection

(b) Pseudo-label approach

(c) Generative Feature Selection

Figure 1: Illustration of different feature selection approaches

Definition 2 (Oracle Affinity) *Oracle affinity is determined by the dot product of oracle features of two nodes.*

$$a_{ij} = \mathbf{x}_i^T diag(\mathbf{s})\mathbf{x}_j \tag{1}$$

We assume that the oracle affinity is determined by oracle features rather than all the original features to avoid redundancy and irrelevance in the high-dimensional input space. Consider a collection of computer science papers on different topics (*e.g.*, machine learning, operating system, database) and citation links between them. Indiscriminative terms, such as *propose*, *related* and *conclusion*, contain little information in determining the essential similarity between two papers. Since two linked papers are more likely to share similar topics than two random papers, informative terms such as *LDA*, *classification* and *database* would be useful in generating the links. Thus, if a feature is highly indicative of the existence of links, it is likely to be an informative and discriminative feature. By recovering the oracle features via exploiting network links, we are able to select a set of discriminative features. To achieve this, we introduce the following generative process:

$$\begin{aligned} p_{ij} =& F_g(a_{ij}) \\ E_{ij} \sim& \text{Bernoulli}(p_{ij}) \end{aligned} \tag{2}$$

where $F_g(\cdot)$ is a function that transforms the oracle affinity a_{ij} to the linkage probability p_{ij}. $F_g(\cdot)$ should be non-decreasing so that a larger affinity would lead to a larger probability of connection. For example, it could be the sigmoid function, *i.e.*, $F_g(a_{ij}) = 1/(1+e^{-a_{ij}})$. We further introduce a bias term $b \in \mathbb{R}$, so $F_g(a_{ij}) = 1/(1+e^{-(a_{ij}+b)})$.

Eq (2) describes the generative process from oracle features to the links in networks. By assuming links are i.i.d, the probability of the whole network given the oracle features is as follows:

$$P(G|\mathbf{s}) = \prod_{(i,j)\in E} p_{ij} \cdot \prod_{(i,j)\notin E} (1-p_{ij}) \tag{3}$$

The negative log-likelihood for generating the network links using $F_g(a_{ij}) = \frac{1}{1+e^{-a_{ij}-b}}$ is the following:

$$\begin{aligned} \mathcal{L}_G =& -\log(P(G|\mathbf{s})) \\ =& -\sum_{(i,j)\in E} \log \frac{1}{1+\exp\left(-\mathbf{x}_i^T \text{diag}(\mathbf{s})\mathbf{x}_j - b\right)} \\ & - \sum_{(i,j)\notin E} \log \frac{\exp\left(-\mathbf{x}_i^T \text{diag}(\mathbf{s})\mathbf{x}_j - b\right)}{1+\exp\left(-\mathbf{x}_i^T \text{diag}(\mathbf{s})\mathbf{x}_j - b\right)} \\ =& \sum_{(i,j)\in V\times V} \log(1+\exp\left(-\mathbf{x}_i^T \text{diag}(\mathbf{s})\mathbf{x}_j - b\right)) \\ & + \sum_{(i,j)\notin E} (\mathbf{x}_i^T \text{diag}(\mathbf{s})\mathbf{x}_j + b) \end{aligned} \tag{4}$$

In real-world applications, network data can be very sparse, *i.e.*, linked node pairs are far less than non-linked node pairs. Due to such imbalanced distribution, $\mathcal{L}_G$ would be dominated by the loss on non-linked node pairs. To address this issue, we under-sample the non-linked node pairs to make their size comparable to the linked node pairs. With downsampling, $\mathcal{L}_G$ is reformulated as follows:

$$\begin{aligned} \mathcal{L}_G =& -\sum_{(i,j)\in E} \log \frac{1}{1+\exp\left(-\mathbf{x}_i^T \text{diag}(\mathbf{s})\mathbf{x}_j - b\right)} \\ & - \sum_{(i,j)\in SN} \log \frac{\exp\left(-\mathbf{x}_i^T \text{diag}(\mathbf{s})\mathbf{x}_j - b\right)}{1+\exp\left(-\mathbf{x}_i^T \text{diag}(\mathbf{s})\mathbf{x}_j - b\right)} \\ =& \sum_{(i,j)\in E\cup SN} \log(1+\exp\left(-\mathbf{x}_i^T \text{diag}(\mathbf{s})\mathbf{x}_j - b\right)) \\ & + \sum_{(i,j)\in SN} (\mathbf{x}_i^T \text{diag}(\mathbf{s})\mathbf{x}_j + b) \end{aligned} \tag{5}$$

where SN denotes the set of sampled non-linked node pairs.

It is worth noting that our link generation approach differs from graph regularization (Gu and Han 2011) (Tang and Liu 2012a) in two important aspects: first, graph regularization usually enforces similarity on linked pairs but fails to utilize information from unlinked pairs; second, graph regularization is usually used on cluster membership/latent factors (as in existing pseudo-label methods) rather than directly on the oracle features. And actually, applying graph regularization on $\text{diag}(\mathbf{s})\mathbf{x}_i$ directly will favor those features that appear indiscriminatively since it fails to penalize features that are frequently shared by unlinked pairs.

Modeling Content Generation

In addition to the linkage information, it is critical to incorporate information from the node content. We assume that each node generates its attributes from the set of oracle features with a mapping function. That is to say, the oracle features can be regarded as a succinct summary of all the features. This intuition can be formalized as follows:

$$\begin{aligned}\boldsymbol{\mu}_i =& F_c(\text{diag}(\mathbf{s})\mathbf{x}_i) \\ \mathbf{x}_i \sim& \mathcal{N}(\boldsymbol{\mu}_i, \sigma^2\mathbf{I}_D)\end{aligned} \tag{6}$$

where $\mathcal{N}$ is the Gaussian distribution and $F_c(\cdot)$ is the function that generates $\mathbf{x}_i$ from the oracle features $\text{diag}(\mathbf{s})\mathbf{x}_i$. There could be different choices for the generation function $F_c(\cdot)$. For simplicity, we use a linear mapping as the generating function.

$$F_c(\text{diag}(\mathbf{s})\mathbf{x}_i) = \mathbf{W}^T\text{diag}(\mathbf{s})\mathbf{x}_i \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{D\times D}$ is a projection matrix that represents all the features using oracle features. Only d rows of $\mathbf{W}$ are non-zero which correspond to the non-zero elements of $\mathbf{s}$. If all the original features could be approximated by the oracle features through $F_c(\cdot)$, the oracle features arguably contain the essential information of the node content.

It is easy to verify that, given fixed $\mathbf{W}$, maximizing the log-likelihood of content generation under Eq (6) is equivalent to minimizing the sum of square error:

$$||\mathbf{X}^T\text{diag}(\mathbf{s})\mathbf{W} - \mathbf{X}^T||_F^2 \tag{8}$$

where $||\cdot||_F$ denotes the Frobenius norm of a matrix. By finding the oracle features that minimize Eq (8), we select the most important features that preserve the information of node attributes in the network data. We also need to impose a norm on $\mathbf{W}$ to control its complexity and avoid overfitting. We choose Frobenius norm for the simplicity of optimization.

$$\mathcal{L}_C = ||\mathbf{X}^T\text{diag}(\mathbf{s})\mathbf{W} - \mathbf{X}^T||_F^2 + \beta||\mathbf{W}||_F^2 \tag{9}$$

Note that other distributions could also be used for modeling feature generation. For example, one can consider Bernoulli distribution if the features are binary.

$$\begin{aligned}\boldsymbol{\mu}_i =& \frac{1}{1+\exp(-F_c(\text{diag}(\mathbf{s})\mathbf{x}_i))} \\ \mathbf{x}_i \sim& \text{Bernoulli}(\boldsymbol{\mu}_i)\end{aligned} \tag{10}$$

where $\boldsymbol{\mu}_i$ determines the probability of occurrence of $\mathbf{x}_i$. It is easy to see that both Eq (6) and Eq (10) are special cases of *Generalized Linear Model* (GLM) with different *link functions*. Eq (6) corresponds to linear regression and Eq (10) corresponds to logistic regression.

Combining Things Together

We have discussed how to generate attributes and links from oracle features in previous sections. Now we put things together and aim to select a set of high-quality features that are optimal considering both content and link generation. Therefore, we aim to minimize the negative log-likelihood on both link and content. By assuming the conditional independence of G and C given b, $\mathbf{s}$ and $\mathbf{W}$, the total negative log-likelihood is as follows:

$$\begin{aligned}\min_{\mathbf{s},b,\mathbf{W}} \quad & -\log P(G,C|\mathbf{s},b,\mathbf{W}) = \mathcal{L}_G + \mathcal{L}_C \\ \text{s.t.} \quad & s_p \in \{0,1\}, \forall p = 1,\dots,D \\ & \sum_{p=1}^{D} s_p = d\end{aligned} \tag{11}$$

Optimization

In this section, we develop a method for performing inference with features and links. The optimization problem in Eq (11) is a '0/1' integer programming problem. To make the optimization tractable, we relax the $0/1$ constraint on $\mathbf{s}$ and only require $\mathbf{s}$ to be a real-valued vector in the range of $[0,1]$. Moreover, we can write the summation constraint $\sum_{p=1}^{D} s_p = d$ in the form of Lagrangian:

$$\begin{aligned}\min_{\mathbf{s},b,\mathbf{W}} \quad & \mathcal{L} = \mathcal{L}_G + \mathcal{L}_C + \lambda||\mathbf{s}||_1 \\ \text{s.t.} \quad & 0 \le s_p \le 1, \forall p = 1,\dots,D\end{aligned} \tag{12}$$

where the sparsity-inducing L_1 norm $||\mathbf{s}||_1$ is equal to $\sum_{p=1}^{D} s_p$, because we enforce $\mathbf{s}$ to be non-negative (*i.e.*, $0 \le s_p \le 1$). The value of s_p can be interpreted as the p-th feature's importance score in generating the content and linkage information. Important features would have scores close to 1 and scores of less useful features tend to shrink towards 0. After obtaining the relaxed solution on $\mathbf{s}$, we can rank all the features by their importance scores and select the top d features as the oracle features.

For the optimization problem in Eq (12), we need to optimize jointly on the selection vector $\mathbf{s}$, bias term b and the projection matrix $\mathbf{W}$. Since Eq (12) is not jointly convex on $\mathbf{s}$, b and $\mathbf{W}$, we adopt an alternating optimization framework to obtain a local optima.

Step 1. Fix $\mathbf{W}$ and optimize Eq (12) over $\mathbf{s}$ and b.

With fixed $\mathbf{W}$, Eq (12) is a convex optimization problem on $\mathbf{s}$ and b. For real-valued $\mathbf{s}$, both $\mathcal{L}_G$ and $\mathcal{L}_C$ is differentiable. For the loss incurred on link structures, the gradient of $\mathcal{L}_G$ with respect to $\mathbf{s}$ can be calculated as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_G}{\partial s_p} =& -\sum_{(i,j)\in E\cup SN} x_{ip}^T x_{jp} \frac{\exp(-\mathbf{x}_i^T\text{diag}(\mathbf{s})\mathbf{x}_j - b)}{1+\exp(-\mathbf{x}_i^T\text{diag}(\mathbf{s})\mathbf{x}_j - b)} \\ & + \sum_{(i,j)\in SN} x_{ip}^T x_{jp}\end{aligned} \tag{13}$$

The gradient of $\mathcal{L}_C$ with respect to $\mathbf{s}$ is the following:

$$\frac{\partial \mathcal{L}_C}{\partial s_p} = 2[\mathbf{X}(\mathbf{X}^T\text{diag}(\mathbf{s})\mathbf{W} - \mathbf{X}^T)\mathbf{W}^T]_{pp} \tag{14}$$

where $[\cdot]_{pp}$ denotes the p-th diagonal element of matrix $[\cdot]$.

The L_1 norm in general is non-smooth at zero. However, since in our case $\mathbf{s}$ is guaranteed to be non-negative, the L_1

regularization on non-negative $\mathbf{s}$ is differentiable with gradient 1. So the gradient of the whole objective function is

$$\frac{\partial \mathcal{L}}{\partial s_p} = \frac{\partial \mathcal{L}_G}{\partial s_p} + \frac{\partial \mathcal{L}_C}{\partial s_p} + \lambda \qquad (15)$$

Since we also require $\mathbf{s}$ to be in the range $[0, 1]$, we perform Projected Gradient Descent (PGD) (Calamai and Moré 1987) for this constrained optimization problem. We project $\mathbf{s}$ back to $[0, 1]$ after each gradient updating step.

$$\text{Proj}_{[0,1]}(s_p) = \min(\max(0, s_p), 1), \forall p = 1, \ldots, D \quad (16)$$

Moreover, the gradient with respect to the bias term b is

$$\frac{\partial \mathcal{L}_G}{\partial b} = - \sum_{(i,j) \in E \cup SN} \frac{\exp(-\mathbf{x}_i^T \text{diag}(\mathbf{s}) \mathbf{x}_j - b)}{1 + \exp(-\mathbf{x}_i^T \text{diag}(\mathbf{s}) \mathbf{x}_j - b)} + |SN| \qquad (17)$$

where $|SN|$ denotes the total number of sampled non-linked node pairs.

Step 2. Fix $\mathbf{s}$ and b, optimize Eq (12) over $\mathbf{W}$.

With fixed $\mathbf{s}$, the optimization with respect to $\mathbf{W}$ is convex and we can obtain the closed form solution for $\mathbf{W}$ as follows:

$$\mathbf{W} = (\text{diag}(\mathbf{s}) \mathbf{X} \mathbf{X}^T \text{diag}(\mathbf{s}) + \beta \mathbf{I}_D)^{-1} \text{diag}(\mathbf{s}) \mathbf{X} \mathbf{X}^T \quad (18)$$

where $\mathbf{I}_D$ is an $D \times D$ identity matrix. Algorithm 1 shows the optimization method based on projected gradient descent. We alternatively perform step 1 and step 2 in an iterative manner until it converges or reaches user-specified maximum number of iterations.

The objective function in Eq (12) monotonically decreases in each iteration and it has a lower bound. Hence, Algorithm 1 can converge to a local minima of the objective (proof can be derived in similar manner as in (Grippo and Sciandrone 2000)).

Algorithm 1 Alternating Optimization with Projected Gradient Descent

```
Initialize: $\mathbf{s}^0 = 0^D$, $b^0 = 0$, $\mathbf{W}^0 = 0^{D \times D}$, $t = 0$.
repeat
    $t = t + 1$
    Update $\mathbf{s}^t$ and $b^t$ through performing projected gradient descent by Eq (15) and Eq (17) with $\mathbf{W}^{(t-1)}$
    Find the optimal $\mathbf{W}^t$ by Eq (18) with $\mathbf{s}^t$.
until converged or $t = maxIterations$
Sort features w.r.t. $\mathbf{s}^t$ and output the top $d$ features.
```

Experiment

In this section, we evaluate the feature quality by performing clustering (community detection) on the features. Experimental results show that GFS significantly outperforms the state-of-the-art methods in terms of feature quality.

Experiment Setup

We use three publicly available network datasets with node attributes: Citeseer dataset, Cora Dataset and Wikipedia dataset [1] (Sen et al. 2008). One can refer to the link in the footnote for more details on the datasets. The statistics of three datasets are summarized in Table 1.

Table 1: Statistics of three datasets

Statistics	Citeseer	Cora	Wiki
# of instances	3312	2708	3363
# of links	4598	5429	33219
# of features	3703	1433	4973
# of classes	6	7	19

We compared our approach to the following baseline methods: (a) All Features; (b) Link Only (Spectral clustering using network links); (c) LS (Laplacian Score) (He, Cai, and Niyogi 2005); (d) UDFS (content only) (Yang et al. 2011) (e) LUFS (which incorporates both content and link information) (Tang and Liu 2012b); (f) RSFS (content only) (Shi, Du, and Shen 2014).

Following the typical setting (Yang et al. 2011) (Tang and Liu 2012b) of evaluation for unsupervised feature selection, we use Accuracy and Normalized Mutual Information (NMI) to evaluate the result of clustering. Accuracy is measured as follows.

$$Accuracy = \frac{1}{n} \sum_{i=1}^{n} \mathcal{I}(c_i = map(p_i)) \qquad (19)$$

where p_i is the clustering result of data point i and c_i is its ground truth label. $map(\cdot)$ is a permutation mapping function that maps p_i to a class label using Kuhn-Munkres Algorithm.

NMI is calculated as follows. Let C be the set of clusters from the ground truth and C' is obtained from a clustering algorithm.

$$NMI(C, C') = \frac{MI(C, C')}{max(H(C), H(C'))} \qquad (20)$$

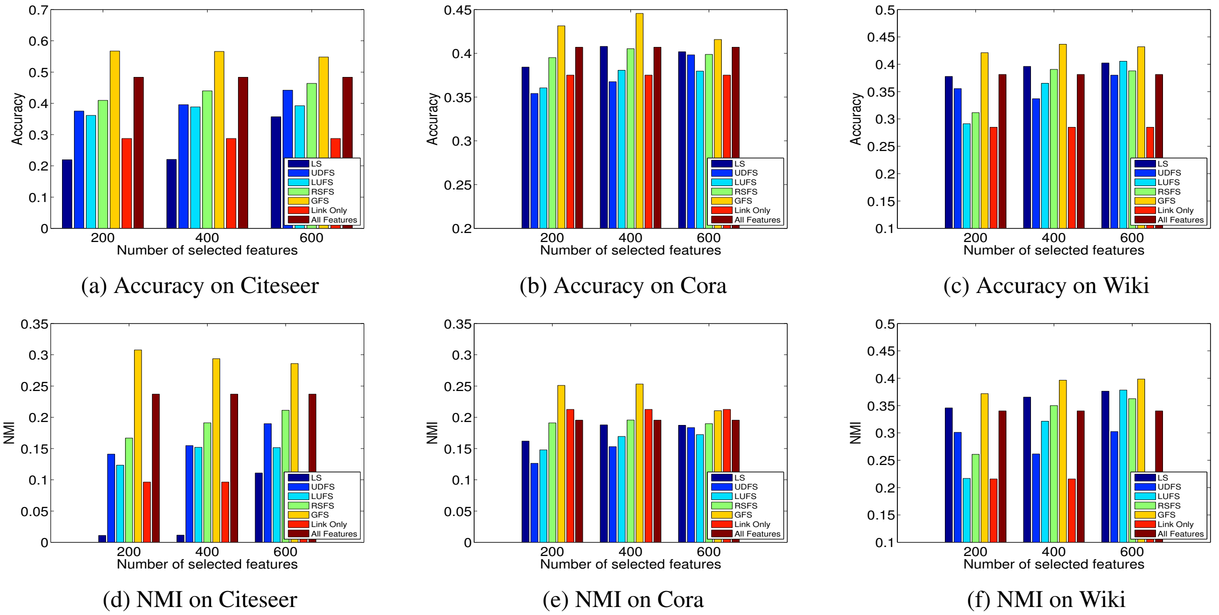
where $H(C)$ and $H(C')$ are the entropy of C and C' and $MI(C, C')$ is the mutual information. Higher value of NMI indicates better quality of clustering.

Since it is difficult to determine the optimal values of parameters in unsupervised setting, we use the parameter setting for the baseline methods as suggested in the sensitivity analysis section of the original papers. For the number of pseudo classes in UDFS, LUFS and RSFS, we use the ground-truth number of classes. For the proposed method GFS, we found it is not sensitive to the parameters in a reasonable range. So we fix the parameters of GFS for all datasets with $\beta = 1$ and $\lambda = 1$.

As in previous work (Yang et al. 2011) (Tang and Liu 2012b), we use K-means[2] for evaluation. Since K-means is affected by the initial seeds, we repeat the experiment for 20 times and report the average performance. We vary the number of features in the range $\{200, 400, 600\}$. The K-means clustering performance for three datasets is shown in Figure 2.

[1] http://linqs.cs.umd.edu/projects/ /projects/lbc/index.html

[2] We use the code at http://www.cad.zju.edu.cn/home/dengcai/Data/Clustering.html



(a) Accuracy on Citeseer

(b) Accuracy on Cora

(c) Accuracy on Wiki

(d) NMI on Citeseer

(e) NMI on Cora

(f) NMI on Wiki

Figure 2: Clustering results on three datasets

Results

We can observe from Figure 2 that feature selection is an effective way to enhance the clustering/community detection performance. With much less features, GFS can obtain significantly better accuracy and NMI than using all the features. For instance, compared with using all features, GFS with 200 features improves the accuracy of clustering by 21.0%, 6.0% and 10.4% on Citeseer, Cora and Wikipedia, respectively. This illustrates the importance of feature selection on networks, since the original feature space can have many low quality/noisy features. It is also worth noting that clustering using only links does not perform very well. This is because network links are often sparse and noisy, and structural information alone is not sufficient to obtain good clusters. But using link structures as guidance in addition to the node content to select features can achieve much better performance, which illustrates the strength of our proposed GFS framework.

When comparing GFS with other unsupervised feature selection approaches, we observe that GFS performs consistently better than baseline methods on different datasets with different numbers of selected features. This indicates that the proposed generative view is an effective framework for selecting high-quality features on network data. LS, UDFS and RSFS are unable to exploit network structure and do not perform as well as GFS. Compared with the most competitive feature selection baseline RSFS, GFS outperforms RSFS by 44.5%, 9.2% and 35.2% with 200 features on three datasets, respectively. Baseline LUFS also attempts to exploit link information via extracting social dimensions (Tang and Liu 2009) from links. But social dimensions extracted from noisy and sparse links can be unreliable and this may further mislead the feature selection process. For example, in Citeseer dataset, the network is sparse and each node only has 1.39 links on average. So the derived social dimensions make LUFS even worse than UDFS and RSFS which do not utilize linkage information. In contrast, GFS can benefit from exploiting the links even when the network structure is sparse, as shown in the case of Citeseer dataset.

In summary, noisy features can be detrimental to the performance of clustering/community detection and appropriately designed unsupervised feature selection method, such as GFS, can alleviate this issue.

Conclusion

In the era of big data, many data instances are connected through link structures in the form of social/information networks. While network links contain valuable information, most state-of-the-art unsupervised feature selection methods either do not utilize the linkage information or utilize the linkage information in a less effective way. In this paper, we develop an unsupervised feature selection algorithm from a generative point of view which can incorporate information from the node content and links directly. We assume that the node attributes and link structures are generated from a set of oracle features and we aim to recover this set of high-quality features based on the generation process. Experiments indicate that our approach significantly outperforms state-of-the-art methods in terms of feature quality.

Acknowledgements

This work is supported in part by NSF through grants III-1526499, CNS-1115234, and OISE-1129076, and Google Research Award.

References

Backstrom, L., and Leskovec, J. 2011. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, 635–644.

Calamai, P., and Moré, J. 1987. Projected gradient methods for linearly constrained problems. *Mathematical Programming* 39:93–116.

Cawley, G. C.; Talbot, N. L. C.; and Girolami, M. 2006. Sparse multinomial logistic regression via bayesian l1 regularisation. In *NIPS*, 209–216.

Dy, J. G., and Brodley, C. E. 2004. Feature selection for unsupervised learning. *Journal of Machine Learning Research* 5:845–889.

Grippo, L., and Sciandrone, M. 2000. On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. *Operations Research Letters* 26:127–136.

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

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

Li, Z.; Yang, Y.; Liu, J.; Zhou, X.; and Lu, H. 2012. Unsupervised feature selection using nonnegative spectral analysis. In *AAAI*.

Newman, M. E. J., and Girvan, M. 2004. Finding and evaluating community structure in networks. *Phys. Rev. E* 69(2):026113.

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

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *NIPS*, 849–856. MIT Press.

Nie, F.; Huang, H.; Cai, X.; and Ding, C. H. Q. 2010. Efficient and robust feature selection via joint l2, 1-norms minimization. In *NIPS*, 1813–1821.

Peng, H.; Long, F.; and Ding, C. H. Q. 2005. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* 27(8):1226–1238.

Qian, M., and Zhai, C. 2013. Robust unsupervised feature selection. In *IJCAI*.

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

Shi, L.; Du, L.; and Shen, Y. 2014. Robust spectral learning for unsupervised feature selection. In *ICDM*, 977–982.

Tang, L., and Liu, H. 2009. Relational learning via latent social dimensions. In *KDD*.

Tang, J., and Liu, H. 2012a. Feature selection with linked data in social media. In *SDM*, 118–128.

Tang, J., and Liu, H. 2012b. Unsupervised feature selection for linked social media data. In *KDD*, 904–912.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)* 58:267–288.

Wei, X.; Xie, S.; and Yu, P. S. 2015. Efficient partial order preserving unsupervised feature selection on networks. In *SDM*.

Yang, Y.; Shen, H. T.; Ma, Z.; Huang, Z.; and Zhou, X. 2011. l2, 1-norm regularized discriminative feature selection for unsupervised learning. In *IJCAI*, 1589–1594.

Zhao, Z., and Liu, H. 2007. Spectral feature selection for supervised and unsupervised learning. In *ICML*, volume 227, 1151–1157.

Zhao, Z.; Wang, L.; and Liu, H. 2010. Efficient spectral feature selection with minimum redundancy. In *AAAI*.