# Robust Unsupervised Feature Selection on Networked Data

Jundong Li* Xia Hu† Liang Wu* Huan Liu*

**Abstract**

Feature selection has shown its effectiveness to prepare high-dimensional data for many data mining and machine learning tasks. Traditional feature selection algorithms are mainly based on the assumption that data instances are independent and identically distributed. However, this assumption is invalid in networked data since instances are not only associated with high dimensional features but also inherently interconnected with each other. In addition, obtaining label information for networked data is time consuming and labor intensive. Without label information to direct feature selection, it is difficult to assess the feature relevance. In contrast to the scarce label information, link information in networks are abundant and could help select relevant features. However, most networked data has a lot of noisy links, resulting in the feature selection algorithms to be less effective. To address the above mentioned issues, we propose a robust unsupervised feature selection framework NetFS for networked data, which embeds the latent representation learning into feature selection. Therefore, content information is able to help mitigate the negative effects from noisy links in learning latent representations, while good latent representations in turn can contribute to extract more meaningful features. In other words, both phases could cooperate and boost each other. Experimental results on real-world datasets demonstrate the effectiveness of the proposed framework.

## 1 Introduction

Networked data encodes pairwise relations among instances in a network. It appears in many domains, such as gene interactions in bioinformatics, hyperlinks between web pages, citations among research papers. In addition to the structural interactions, networked instances are often accompanied with high-dimensional features. For example, in Facebook and Twitter, millions of posts are generated every day which bring about high-dimensional feature space. The high-dimensional feature space of networked data poses challenges to many learning tasks due to the curse of dimensionality [6]. Besides, data with high dimensionality significantly increases the memory storage requirements and computational costs for data analytics. Moreover, the existence of irrelevant, redundant and noisy features overfits the learning algorithms and results in low efficiency and poor performance.

Feature selection [21], as a data preprocessing step, has shown to be effective in preparing high-dimensional data for many data mining tasks such as sentiment analysis [14, 32] and node classification [10]. According to the availability of labels, feature selection methods consist of supervised methods [5, 25, 31] and unsupervised methods [3, 12, 19, 34]. As it is easy to amass substantial amounts of unlabeled data while label information is costly to obtain, unsupervised feature selection has received increasingly attention in the past few years. Without label information to help assess feature relevance, unsupervised feature selection algorithms exploit different criteria to define the relevance of features such as data similarity [3, 12, 34], local discriminative information [19, 33] and data reconstruction error [7].

Existing unsupervised feature selection algorithms cannot be directly applied or are not suitable for networked data because of its distinct characteristics: (1) in networks, data instances are not independent and identically distributed (i.i.d.) but inherently interconnected with each other. Meanwhile, they are often associated with some content features. As indicated by homophily effect [23] in social sciences theories, network structure and content information correlates with each other and the formation of one is dependent on the other one; (2) in addition to noisy features in the content space, link information is prevalent and also contains a lot of noise. For example, in social networks, it is easy for spammers to imitate normal users by generating a large number of noisy links; in bank transaction networks, abnormal financial activities also bring some noisy links. These noisy links may have a negative effect in finding relevant features.

Considering the unique properties of networked data mentioned above, we propose a robust unsupervised feature selection framework NetFS. First, to capture the inherent interactions among networked instances, we introduce the concept of latent representations to uncover some hidden attributes encoded in

*Computer Science and Engineering, Arizona State University, Tempe, AZ, USA. {jundongl, wuliang, huan.liu}@asu.edu

†Department of Computer Science and Engineering, Texas A&M University, College Station, TX, USA. hu@cse.tamu.edu

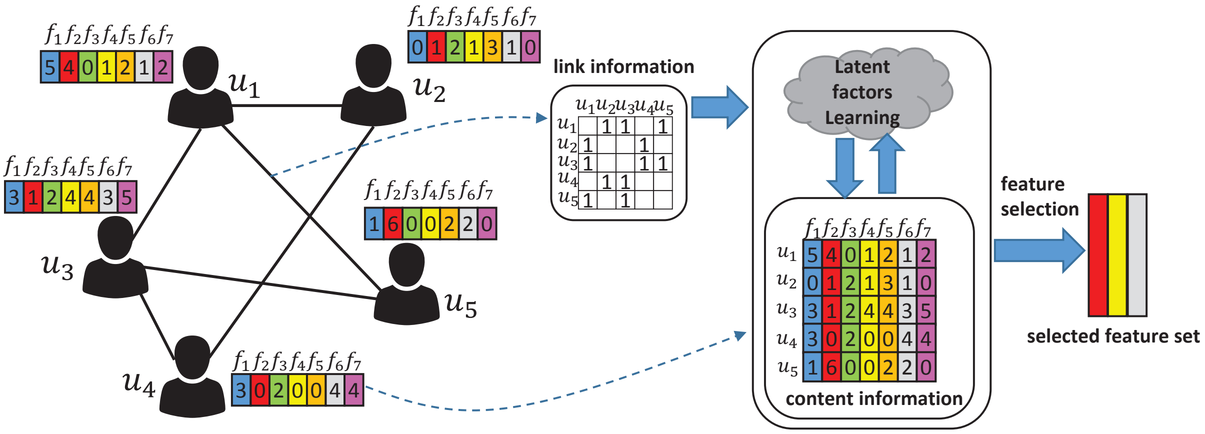


Figure 1: The proposed robust unsupervised feature selection framework NetFS for networked data. It embeds the latent representation learning into the feature selection phase. Each color bar denotes a feature, the values in the color bar show the value of the feature, for example the word frequency as indicated in the figure. Network information is encoded in an adjacency matrix, where the value of 1 indicates the existence of a relationship.

the network structure. Second, to reduce the negative effects from noisy links in feature selection, we propose to embed the latent representation learning into the feature selection phase. In this way, the latent representation modeling could cooperate with the feature selection phase. Specifically, content information helps mitigate the negative effects of noisy links in learning latent representations, the learnt latent representations can not only characterize the network structure but also in turn act as label information to guide the feature selection phase. As a result, the feature selection performance would be better. The main contributions of this paper are as follows:

- Proposing a principled way to embed the latent representation learning from network structure into feature selection phase;
- Proposing a robust unsupervised feature selection framework NetFS for networked data;
- Providing an alternating algorithm to optimize the proposed NetFS framework; and
- Evaluating the efficacy of the proposed NetFS framework on real-world datasets.

The rest of this paper is structured as follows. The problem statement is introduced in Section 2. In Section 3, we talk about the proposed NetFS framework with an optimization method and its convergence analysis. Experimental evaluation on real-world datasets is presented in Section 4 with discussions. In Section 5, we briefly review related work on traditional feature selection methods and feature selection methods for networked data. The conclusions and future work are presented in Section 6.

## 2 Problem Statement

We first summarize some notations used in this paper. Following the commonly used notations, we use bold uppercase characters for matrices (e.g. $\mathbf{A}$), bold lowercase characters for vectors (e.g. $\mathbf{a}$), normal lowercase characters for scalars (e.g. $a$), calligraphic fonts for sets (e.g. $\mathcal{F}$). Also, we follow the matrix settings in matlab to represent $i$-th row of matrix $\mathbf{A}$ as $\mathbf{A}(i,:)$, $j$-th column as $\mathbf{A}(:,j)$, $(i,j)$-th entry as $\mathbf{A}(i,j)$, transpose as $\mathbf{A}'$, trace as $Tr(\mathbf{A})$ if $\mathbf{A}$ is a square matrix. For any matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, its Frobenius norm is defined as $||\mathbf{A}||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{d}\mathbf{A}(i,j)^2}$, its $\ell_{2,1}$-norm is $||\mathbf{A}||_{2,1} = \sum_{i=1}^{n}\sqrt{\sum_{j=1}^{d}\mathbf{A}(i,j)^2}$. $\mathbf{I}_n$ denotes the identity matrix of size $n$.

Let $\mathcal{U} = \{u_1, u_2, ..., u_n\}$ denote a set of $n$ linked instances in the network, we use the adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ to represent the network structure of $\mathcal{U}$. For undirected network, the adjacency matrix is symmetric such that $\mathbf{A} = \mathbf{A}'$. To model the network information on directed networks, we follow [8] to use $\mathbf{A} = max(\mathbf{A}, \mathbf{A}')$. Each linked instance is associated with a set of $d$ features $\mathcal{F} = \{f_1, f_2, ..., f_d\}$, $\mathbf{X} \in \mathbb{R}^{n\times d}$ denotes the content information of all $n$ instances. With above notations, the problem statement is as follows:

Problem 1. *Unsupervised Feature Selection for Networked Data*

**Given:** *The feature set $\mathcal{F}$, content matrix $\mathbf{X}$ and adjacency matrix $\mathbf{A}$ for all $n$ instances.*

**Select:** *A subset of most relevant features $\mathcal{S} \subseteq \mathcal{F}$ by exploiting both content information* $\mathbf{X}$ *and network information* $\mathbf{A}$.

## 3 Robust Unsupervised Feature Selection for Networked Data - NetFS

In this section, we introduce the proposed NetFS framework which embeds latent representation learning into feature selection in detail.

The workflow of the proposed framework is illustrated in Figure 1. From the figure, we can see that we have two sources of information, i.e., the content information and the link information. At the first step, we try to model the link information with the concept of latent representations. In the second step, we embed the latent representation learning into feature selection. Actually, these two steps influence and help each other iteratively. The output of NetFS is a subset of relevant features in the content space. We first discuss how to model the latent representations from network structure, and then introduce how to embed the latent representation learning into content information for feature selection. At last, we will provide an alternating optimization algorithm for the proposed framework with its convergence analysis.

**3.1 Modeling Link Information with Latent Representation** As illustrated in Figure 1, although link information is abundant, it cannot be directly applied as data is not i.i.d.. In this subsection, we talk about how to exploit link information effectively.

In networks, instances connect to each other due to a variety of factors. For example, in social networks, these factors can be movie fans, sports enthusiasts, colleagues, family members, etc; in coauthor networks, factors include similar research interests, same affiliations, etc. These hidden factors are often referred as latent representations since they can describe a set of diverse affiliation factors hidden in a network. Latent representations of different instances interact with each other to form link information, and the instances with similar latent representations are more likely to be connected with each other than the instances with dissimilar latent representations.

Uncovering latent representations has received increasingly attention recently in the data mining and machine learning communities [1, 24, 30]. Here, we model the latent representations from link information by a symmetric nonnegative matrix factorization model (SymNMF) [13, 16]. The principle of SymNMF is consistent with network clustering such that each networked instance consists of a mixture of latent attributes. Mathematically, it decomposes the adjacency matrix $\mathbf{A}$ into a product of a nonnegative matrix $\mathbf{U}$ and its transpose $\mathbf{U}'$ in a low-dimensional latent space:

$$\min_{\mathbf{U}\geq 0} ||\mathbf{A} - \mathbf{U}\mathbf{U}'||_F^2, \tag{3.1}$$

where $\mathbf{U} \in \mathbb{R}^{n\times c}$ is the latent representations of all $n$ instances, $c$ is the number of latent factors.

**3.2 Embedding Latent Representation Learning into Feature Selection** It can be observed from Figure 1, after we model the network structure by latent representations, we can take advantage of them for feature selection. In this section, we introduce how to perform feature selection in a robust manner.

With the existence of noisy links in networked data, latent representations that are directly derived from link information may jeopardize feature selection on the content space. In addition, according to the homophily effect [23] in network studies, content information will affect and is dependent on the latent representations from network structure. Therefore, it is desirable to embed the latent representation learning into the feature selection phase on the content space. As a result, the latent representation learning and feature selection could help and boost each other. Content information can help learn better latent representations which are robust to noisy links, and better latent representations can fill the gap of scarce label information and rich link information to guide feature selection.

As latent factors encode some hidden attributes of instances, they should be related to some features (or attributes) of networked instances. Therefore, we take $\mathbf{U}$ as a constraint to model the content information through a multivariate linear regression model:

$$\min_{\mathbf{W}} ||\mathbf{X}\mathbf{W} - \mathbf{U}||_F^2, \tag{3.2}$$

where $\mathbf{W} \in \mathbb{R}^{d\times c}$ is a transformation parameter matrix, each row vector $\mathbf{W}(i,:)$ measures the importance of the $i$-th feature. To achieve feature selection, we add a $\ell_{2,1}$-norm regularization term on $\mathbf{W}$ for a joint sparsity among all $c$ latent factors:

$$\min_{\mathbf{W}} ||\mathbf{X}\mathbf{W} - \mathbf{U}||_F^2 + \alpha||\mathbf{W}||_{2,1}, \tag{3.3}$$

where parameter $\alpha$ controls the sparseness of the model.

By combining the objective function in Eq. (3.1) and Eq. (3.3), the objective function that embeds latent representation learning into feature selection phase can be formulated as follows:

$$\begin{aligned}\min_{\mathbf{U}\geq 0,\mathbf{W}} \mathcal{J}(\mathbf{W},\mathbf{U}) &= ||\mathbf{X}\mathbf{W} - \mathbf{U}||_F^2 + \alpha||\mathbf{W}||_{2,1} \\ &+ \frac{\beta}{2}||\mathbf{A} - \mathbf{U}\mathbf{U}'||_F^2,\end{aligned} \tag{3.4}$$

where $\beta$ is a parameter to balance the latent representation modeling and the feature selection in content space. It can be observed from Eq. (3.4) than when $\mathbf{W}$ is fixed, the latent representation learning phase is not only associated with the adjacency matrix $\mathbf{A}$, but also the content matrix $\mathbf{X}$. In this way, the learnt latent representations can capture their inherent correlations and are more robust to noisy links. When the latent representations $\mathbf{U}$ is fixed, they will take the role of label information to steer feature selection in a supervised way.

**3.3 Optimization Solution** In this subsection, we talk about how to solve the optimization problem of NetFS. The objective function in Eq. (3.4) is not convex in both $\mathbf{U}$ and $\mathbf{W}$ simultaneously, besides, due to the $\ell_{2,1}$-norm regularization term, it is also not smooth. Motivated by [25], we adopt an alternating optimization scheme to solve this problem.

When $\mathbf{U}$ is fixed, the objective function is convex w.r.t. $\mathbf{W}$, therefore, we take the derivative of $\mathcal{J}(\mathbf{W},\mathbf{U})$ with respect to $\mathbf{W}$ to be zero, then we have:

$$\mathbf{X}'(\mathbf{X}\mathbf{W}-\mathbf{U})+\alpha\mathbf{D}\mathbf{W}=0, \tag{3.5}$$

where $\mathbf{D}\in\mathbb{R}^{d\times d}$ is a diagonal matrix, with the $i$-th diagonal element as:

$$\mathbf{D}(i,i)=\frac{1}{2||\mathbf{W}(i,:)||_2}. \tag{3.6}$$

It should be noted that in practice, $||\mathbf{W}(i,:)||_2$ could be very close to zero. Therefore, we define $\mathbf{D}(i,i)=\frac{1}{2||\mathbf{W}(i,:)||_2+\epsilon}$, where $\epsilon$ is a very small constant. $\mathbf{X}'\mathbf{X}$ is a positive semi-definite matrix, $\alpha\mathbf{D}$ is a diagonal matrix with positive entries, it is also positive semi-definite. Their summation $\mathbf{X}'\mathbf{X}+\alpha\mathbf{D}$ is also positive semi-definite. Therefore, $\mathbf{W}$ has a closed form solution, which is:

$$\mathbf{W}=(\mathbf{X}'\mathbf{X}+\alpha\mathbf{D})^{-1}\mathbf{X}'\mathbf{U}. \tag{3.7}$$

By substituting above solution of $\mathbf{W}$ into Eq. (3.4), we get:

$$\begin{aligned}
&\min_{\mathbf{U}\geq 0,\mathbf{W}} \mathcal{J}(\mathbf{W},\mathbf{U})\\
=&Tr(\mathbf{U}'\mathbf{U})-Tr(\mathbf{W}'\mathbf{M}\mathbf{W})+\frac{\beta}{2}||\mathbf{A}-\mathbf{U}\mathbf{U}'||_F^2\\
=&Tr(\mathbf{U}'\mathbf{U})-Tr(\mathbf{U}'\mathbf{X}\mathbf{M}^{-1}\mathbf{X}'\mathbf{U})+\frac{\beta}{2}||\mathbf{A}-\mathbf{U}\mathbf{U}'||_F^2\\
=&Tr(\mathbf{U}'(\mathbf{I}_n-\mathbf{X}\mathbf{M}^{-1}\mathbf{X}')\mathbf{U})+\frac{\beta}{2}||\mathbf{A}-\mathbf{U}\mathbf{U}'||_F^2,
\end{aligned} \tag{3.8}$$

where $\mathbf{M}=\mathbf{X}'\mathbf{X}+\alpha\mathbf{D}$.

The problem in Eq. (3.8) is a standard bound-constrained optimization problem, we propose to use projected gradient methods [20] to optimize it. Now the objective function can be reformulated as:

$$\min_{\mathbf{U}\geq 0}\mathcal{J}(\mathbf{U})=Tr(\mathbf{U}'(\mathbf{I}_n-\mathbf{X}\mathbf{M}^{-1}\mathbf{X}')\mathbf{U})+\frac{\beta}{2}||\mathbf{A}-\mathbf{U}\mathbf{U}'||_F^2. \tag{3.9}$$

Let $\mathbf{U}_t$ denotes the update of $\mathbf{U}$ at the $t$-th iteration. It is updated by the following rule:

$$\mathbf{U}_{t+1}=P[\mathbf{U}_t-s_t\nabla\mathcal{J}(\mathbf{U}_t)], \tag{3.10}$$

where $P[\mathbf{U}_t-s_t\nabla\mathcal{J}(\mathbf{U}_t)]$ is a box projection operator which maps a point to a bounded region, let $\mathbf{C}=\mathbf{U}_t-s_t\nabla\mathcal{J}(\mathbf{U}_t)$, then:

$$P[\mathbf{C}](i,j)=\begin{cases}\mathbf{C}(i,j) & \text{if } \mathbf{C}(i,j)\geq 0\\ 0 & \text{otherwise}\end{cases} \tag{3.11}$$

$s_t$ is the step size at the $t$-th iteration and can be determined by Armijo rule [2]. To be more specific, $s_t=\theta^{a_t}$, where $a_t$ is the first nonnegative integer such that the condition:

$$\mathcal{J}(\mathbf{U}_{t+1})-\mathcal{J}(\mathbf{U}_t)\leq\sigma\langle\nabla\mathcal{J}(\mathbf{U}_t),(\mathbf{U}_{t+1}-\mathbf{U}_t)\rangle \tag{3.12}$$

is satisfied, where $\theta$ and $\sigma$ are two predefined parameters between 0 and 1, $\langle\mathbf{A},\mathbf{B}\rangle$ represents the inner product operation between two matrix $\mathbf{A}$ and $\mathbf{B}$.

In summary, the objective function in Eq. (3.4) is solved through an iterative way as illustrated in Algorithm 1, when $\mathbf{W}$ is fixed, we use projected gradient method through Eq. (3.10) to update $\mathbf{U}$, the computation cost is $O(n^2d)+O(nd^2)+O(n^2c)$; then we fix $\mathbf{U}$ and employ Eq. (3.7) to update $\mathbf{W}$, the total cost of computing $\mathbf{W}$ in Eq. (3.7) is $O(nd^2)+O(d^3)+O(d^2c)+O(ncd)$.

**3.4 Convergence Analysis** We show the objective function in Eq. (3.4) decreases each iteration and it is guaranteed to converge.

LEMMA 3.1. *The following inequality holds if $\mathbf{W}_k(i,:)$ and $\mathbf{W}_{k+1}(i,:)$ are non-zero vectors (i=1,2,...,d) [25]:*

$$\begin{aligned}
&||\mathbf{W}_{k+1}||_{2,1}-\sum_i\frac{||\mathbf{W}_{k+1}(i,:)||_2^2}{2||\mathbf{W}_k(i,:)||_2}\\
\leq&||\mathbf{W}_k||_{2,1}-\sum_i\frac{||\mathbf{W}_k(i,:)||_2^2}{2||\mathbf{W}_k(i,:)||_2}
\end{aligned} \tag{3.13}$$

THEOREM 3.1. *The alternating procedure in Algorithm 1 will decrease the objective function value of Eq. (3.4).*

**Algorithm 1** NetFS algorithm

```
Input: Data matrix $\mathbf{X}$ and adjacency matrix $\mathbf{A}$, parameter $\alpha$, $\beta$.
Output: The most $m$ relevant features.
Initialize $\mathbf{D}_k$ as an identity matrix;
Set $k = 0$;
while objective function value in Eq. (3.4) not converge do
    Compute $\mathbf{M}_k = \mathbf{X}'\mathbf{X} + \alpha\mathbf{D}_k$;
    Obtain $\mathbf{U}_{k+1}$ by projected gradient method in Eq. (3.10);
    Obtain $\mathbf{W}_{k+1}$ by Eq. (3.7);
    Update $\mathbf{D}_{k+1}$ by Eq. (3.6);
    $k = k + 1$;
end while
Sort each feature according to $||\mathbf{W}(i,:)||_2$ in a descending order and select the top-$m$ ranked ones
Return $m$ most relevant features.
```

*Proof.* During the $k+1$-th iteration, when $\mathbf{W}_k$ is fixed, we update $\mathbf{U}$ with projected gradient method. Projected gradient descent method decreases the objective function $\mathcal{J}(\mathbf{U})$ for appropriate choices of step size. Therefore, we have:

$$\mathcal{J}(\mathbf{U}_{k+1}, \mathbf{W}_k) \leq \mathcal{J}(\mathbf{U}_k, \mathbf{W}_k). \tag{3.14}$$

Then when $\mathbf{U}_{k+1}$ is fixed, we obtain the optimal solution $\mathbf{W}_{k+1}$ through Eq. (3.7) and $\mathbf{W}_{k+1}$ is the solution of the following objective function:

$$\min_{\mathbf{W}} ||\mathbf{X}\mathbf{W} - \mathbf{U}_{k+1}||_F^2 + \alpha Tr(\mathbf{W}'\mathbf{D}_k\mathbf{W}) \tag{3.15}$$

Therefore, we have the following inequality:

$$\begin{aligned} & ||\mathbf{X}\mathbf{W}_{k+1} - \mathbf{U}_{k+1}||_F^2 + \alpha Tr(\mathbf{W}'_{k+1}\mathbf{D}_k\mathbf{W}_{k+1}) \\ \leq\ & ||\mathbf{X}\mathbf{W}_k - \mathbf{U}_{k+1}||_F^2 + \alpha Tr(\mathbf{W}'_k\mathbf{D}_k\mathbf{W}_k) \\ \Rightarrow\ & ||\mathbf{X}\mathbf{W}_{k+1} - \mathbf{U}_{k+1}||_F^2 + \alpha||\mathbf{W}_{k+1}||_{2,1} \\ & - \alpha(||\mathbf{W}_{k+1}||_{2,1} - \sum_i \frac{||\mathbf{W}_{k+1}(i,:)||_2^2}{2||\mathbf{W}_k(i,:)||_2}) \\ \leq\ & ||\mathbf{X}\mathbf{W}_k - \mathbf{U}_{k+1}||_F^2 + \alpha||\mathbf{W}_k||_{2,1} \\ & - \alpha(||\mathbf{W}_k||_{2,1} - \sum_i \frac{||\mathbf{W}_k(i,:)||_2^2}{2||\mathbf{W}_k(i,:)||_2}) \end{aligned} \tag{3.16}$$

Integrating the results in Lemma 3.1, we have the following:

$$\begin{aligned} & ||\mathbf{X}\mathbf{W}_{k+1} - \mathbf{U}_{k+1}||_F^2 + \alpha||\mathbf{W}_{k+1}||_{2,1} \\ \leq\ & ||\mathbf{X}\mathbf{W}_k - \mathbf{U}_{k+1}||_F^2 + \alpha||\mathbf{W}_k||_{2,1} \\ \Rightarrow\ & \mathcal{J}(\mathbf{U}_{k+1}, \mathbf{W}_{k+1}) \leq \mathcal{J}(\mathbf{U}_{k+1}, \mathbf{W}_k) \leq \mathcal{J}(\mathbf{U}_k, \mathbf{W}_k) \end{aligned} \tag{3.17}$$

which completes the proof.

| | BlogCatalog | Flickr | Epinions |
|---|---|---|---|
| # of Users | 5,196 | 7,575 | 5,665 |
| # of Features | 8,189 | 12,047 | 10,382 |
| # of Links | 171,743 | 239,738 | 97,123 |
| # of Ave Degree | 66.11 | 63.30 | 17.14 |
| # of Classes | 6 | 9 | 24 |

Table 1: Detailed information of the datasets.

## 4 Experiments

In this section, we conduct experiments to evaluate the effectiveness of the proposed NetFS framework. We first introduce the datasets and experimental settings before presenting details of the experiments.

### 4.1 Datasets

Three real-world social media datasets, BlogCatalog[1], Flickr[1] and Epinions[2] are used for evaluation. Some statistics of these datasets are listed in Table 1.

**BlogCatalog:** BlogCatalog is a social blog directory in which users can post their blogs under different predefined categories. The tag information of blogs from users form the feature information, while the major categories of blogs by users are considered as ground truth in this work.

**Flickr:** Flickr is an image sharing website, users can provide tags for the photos they upload which provide feature information. Besides, users interact with others forming link information. Photos are organized under some predefined categories, which are used as ground truth.

**Epinions:** Epinions is a product review website in which users can share their reviews about products. Users themselves can also build trust networks to seek advices from others. Features are formed by the bag-of-words model, while the major categories of reviews by users are taken as ground truth of class labels.

### 4.2 Experimental Settings

Following the standard way to assess unsupervised feature selection [3, 19, 33], we evaluate the proposed NetFS framework in terms of clustering performance. Two commonly used clustering performance metrics, i.e., *normalized mutual information* (NMI) and *accuracy* (ACC) are used. Let $C$ and $C'$ denote the clustering results from ground truth class labels and the predicted cluster labels, respectively. The mutual information between two

[1] http://dmml.asu.edu/users/xufei/datasets.html

[2] http://jiliang.xyz/trust.html

clusters $C$ and $C'$ is:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) log \frac{p(c_i, c'_j)}{p(c_i)p(c'_j)} \tag{4.18}$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities of instances in cluster $c_i$ and $c'_j$, respectively. $p(c_i, c'_j)$ indicates the probability of instances in cluster $c_i$ and in $c'_j$ at the same time. Then, NMI is defined as:

$$NMI(C, C') = \frac{MI(C, C')}{max(H(C), H(C')} \tag{4.19}$$

where $H(C)$ and $H(C')$ represent the entropies of clusterings $C$ and $C'$, respectively.

Let $p_i$ and $q_i$ be the clustering result and the ground truth label for instance $u_i$, respectively. Then, accuracy (ACC) is defined as:

$$ACC = \frac{1}{n} \sum_{i=1}^{n} \delta(q_i, map(p_i)) \tag{4.20}$$

where $n$ is the total number of instances, $\delta(.)$ is an indicator function such that $\delta(x, y) = 1$ if $x = y$, otherwise $\delta(x, y) = 0$. $map(x)$ permutes the predicted cluster labels to match the ground truth as much as possible.

NetFS is measured against the following state-of-the-art unsupervised feature selection algorithms in terms of clustering performance:

- **LapScore**: Laplacian score [12] evaluates feature via its ability of locality preservation.
- **SPEC**: SPEC is an extension of Laplacian Score, features are selected using spectral analysis [34].
- **NDFS**: Features are selected via joint nonnegative spectral analysis and $\ell_{2,1}$-norm regularization [19].
- **LUFS**: Social dimensions are first extracted from link information, then they are utilized to guide feature selection in the content space [29].

In LapScore and NDFS, as suggested by the original papers [12, 19], we set the number of neighborhood size to be 5 to construct the affinity matrix. NDFS and LUFS have different regularization parameters, we set these parameters by the suggestions from the original papers [19, 29]. In the proposed NetFS, we also have two regularization parameters $\alpha$ and $\beta$. These two parameters can be determined by grid search strategy among the range of $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. In the experiments, we empirically set $\alpha$ as 10 and $\beta$ as 0.1.

Each feature selection algorithm is first applied to select features, then K-means clustering is performed based on the selected features. We repeat the K-means algorithm 20 times and report the average clustering results since K-means may converge to local optimal.

**4.3 Quality of Selected Features by NetFS** In this subsection, we compare the quality of selected features by NetFS and other baseline methods on the three above mentioned datasets. The number of selected features varies as $\{200, 400, ..., 1800, 2000\}$. The comparison results are shown in Table 2, Table 3 and Table 4. The higher the ACC and NMI values are, the better the feature selection performance is. We make the following observations:

- NetFS outperforms traditional unsupervised feature selection algorithms in almost all cases by obtaining better clustering performance. We also perform pairwise Wilcoxon signed-rank test [4, 18] between NetFS and these baseline methods, the test results show NetFS is significantly better, with a 0.05 significance level. A major reason is that traditional algorithms can only handle *i.i.d.* data while NetFS exploits content information and network structure to obtain good features.
- NetFS and LUFS deal with network and content information differently. LUFS performs network structure modeling and feature selection separately and the feature selection performance is highly dependent on the quality of extracted latent representations; and the proposed NetFS embeds the latent representation learning phase into the feature selection, therefore, content information is used adaptively to obtain better latent factor representations because better latent factors can contribute to selecting more relevant features.
- In Blogcatalog dataset, NetFS works well with only a few hundred of features. Flickr and Epinions datasets have more features than Blogcatalog, but NetFS still achieves good clustering performance with only around 1/8 and 1/7 of total features, respectively.

**4.4 Effect of Parameters** NetFS has two regularization parameters $\alpha$ and $\beta$. $\alpha$ controls the sparsity of the model while $\beta$ balances the latent representation learning and feature selection phase. To investigate the effects of these two parameters, we fix one parameter each time and vary the other one to see how it affects the feature selection performance. As the settings mentioned above, we assess the feature selection performance in terms of clustering with different number of selected features. In Figure 2(a), we present the clustering performance of NetFS on Epinions dataset in terms of ACC. We only report the parameter study results of ACC on Epinions dataset to save space as we have the similar observations in terms of NMI.

| | Accuracy | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 200 | 400 | 600 | 800 | 1000 | 1200 | 1400 | 1600 | 1800 | 2000 |
| LapScore | 26.75 | 28.95 | 26.35 | 24.52 | 32.91 | 27.66 | 27.96 | 29.94 | 30.75 | 32.37 |
| SPEC | 17.90 | 18.01 | 18.90 | 19.55 | 20.52 | 23.73 | 22.32 | 21.11 | 26.00 | 24.23 |
| NDFS | 24.61 | 32.35 | 33.43 | 31.89 | 34.85 | 32.99 | 33.76 | **45.57** | 42.90 | 43.15 |
| LUFS | 21.52 | 21.70 | 31.72 | 31.79 | 32.39 | 32.60 | 34.16 | 42.13 | 41.28 | 43.49 |
| NetFS | **50.89** | **42.38** | **42.96** | **42.73** | **43.17** | **43.30** | **43.36** | 43.61 | **43.61** | **43.65** |
| | NMI | | | | | | | | | |
| | 200 | 400 | 600 | 800 | 1000 | 1200 | 1400 | 1600 | 1800 | 2000 |
| LapScore | 0.0703 | 0.0720 | 0.0557 | 0.0386 | 0.0836 | 0.0465 | 0.0460 | 0.0577 | 0.0622 | 0.0703 |
| SPEC | 0.0016 | 0.0018 | 0.0052 | 0.0073 | 0.0083 | 0.0222 | 0.0246 | 0.0181 | 0.0493 | 0.0341 |
| NDFS | 0.1053 | 0.1597 | 0.1925 | 0.1381 | 0.1699 | 0.1466 | 0.1531 | 0.2298 | 0.2329 | 0.2356 |
| LUFS | 0.0374 | 0.0397 | 0.1417 | 0.1398 | 0.1646 | 0.1739 | 0.1946 | **0.2751** | **0.2677** | 0.2328 |
| NetFS | **0.3264** | **0.2276** | **0.2363** | **0.2345** | **0.2349** | **0.2374** | **0.2378** | 0.2409 | 0.2401 | **0.2403** |

Table 2: Clustering results with different feature selection algorithms on BlogCatalog dataset.

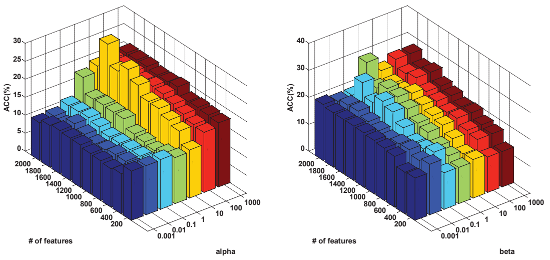
| | Accuracy | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 200 | 400 | 600 | 800 | 1000 | 1200 | 1400 | 1600 | 1800 | 2000 |
| LapScore | 12.30 | 12.37 | 12.42 | 13.21 | 13.28 | 13.65 | 14.96 | 16.03 | 17.04 | 16.86 |
| SPEC | 11.87 | 12.48 | 13.15 | 13.89 | 14.32 | 14.27 | 14.67 | 15.33 | 14.73 | 14.86 |
| NDFS | 15.52 | 17.23 | 27.21 | 29.94 | **35.70** | 33.89 | 37.65 | 39.42 | 42.06 | **46.43** |
| LUFS | 13.25 | 18.19 | 20.40 | 23.59 | 22.53 | 30.07 | 27.99 | 29.16 | 35.22 | 39.51 |
| NetFS | **22.18** | **30.39** | **35.49** | **36.51** | 35.52 | **43.29** | **45.35** | **40.29** | **48.17** | 36.21 |
| | NMI | | | | | | | | | |
| | 200 | 400 | 600 | 800 | 1000 | 1200 | 1400 | 1600 | 1800 | 2000 |
| LapScore | 0.0058 | 0.0069 | 0.0068 | 0.0120 | 0.0194 | 0.0210 | 0.0272 | 0.0384 | 0.0428 | 0.0364 |
| SPEC | 0.0018 | 0.0057 | 0.0077 | 0.0119 | 0.0157 | 0.0139 | 0.0152 | 0.0172 | 0.0160 | 0.0188 |
| NDFS | 0.0327 | 0.0390 | 0.0924 | 0.1146 | 0.1752 | 0.1424 | 0.1942 | 0.2254 | 0.2338 | **0.3000** |
| LUFS | 0.0147 | 0.0687 | 0.0949 | 0.1258 | 0.1051 | 0.1475 | 0.1356 | 0.1491 | 0.2031 | 0.2404 |
| NetFS | **0.1017** | **0.1648** | **0.1825** | **0.1950** | **0.2195** | **0.2285** | **0.2776** | **0.2474** | **0.2935** | 0.2263 |

Table 3: Clustering results with different feature selection algorithms on Flickr dataset.

| | Accuracy | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 200 | 400 | 600 | 800 | 1000 | 1200 | 1400 | 1600 | 1800 | 2000 |
| LapScore | **15.22** | 13.35 | 12.44 | 11.95 | 11.63 | 11.88 | 11.84 | 11.58 | 11.30 | 11.23 |
| SPEC | 14.67 | 12.10 | 11.09 | 10.95 | 10.89 | 10.47 | 10.87 | 11.37 | 11.53 | 10.73 |
| NDFS | 12.60 | 11.97 | 12.34 | 12.68 | 12.79 | 14.69 | 14.16 | 15.80 | 14.28 | 15.43 |
| LUFS | 12.70 | 11.48 | 11.14 | 11.81 | 12.87 | 14.12 | 13.24 | 14.21 | 13.65 | 17.19 |
| NetFS | 13.93 | **16.77** | **18.45** | **20.56** | **20.72** | **21.78** | **24.61** | **23.83** | **24.40** | **27.17** |
| | NMI | | | | | | | | | |
| | 200 | 400 | 600 | 800 | 1000 | 1200 | 1400 | 1600 | 1800 | 2000 |
| LapScore | 0.0146 | 0.0197 | 0.0211 | 0.0217 | 0.0213 | 0.0213 | 0.0226 | 0.0230 | 0.0210 | 0.0203 |
| SPEC | 0.0170 | 0.0219 | 0.0245 | 0.0256 | 0.0263 | 0.0253 | 0.0273 | 0.0286 | 0.0278 | 0.0259 |
| NDFS | 0.0234 | 0.0264 | 0.0291 | 0.0320 | 0.0348 | 0.0395 | 0.0436 | 0.0464 | 0.0494 | 0.0552 |
| LUFS | 0.0212 | 0.0232 | 0.0246 | 0.0280 | 0.0325 | 0.0375 | 0.0392 | 0.0418 | 0.0446 | 0.0501 |
| NetFS | **0.0359** | **0.0547** | **0.0661** | **0.0746** | **0.0964** | **0.1020** | **0.1106** | **0.1124** | **0.1131** | **0.0808** |

Table 4: Clustering results with different feature selection algorithms on Epinions dataset.

We first fix the parameter $\beta$ to be 0.1 and vary the other parameter $\alpha$ as $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. As shown in Figure 2(a), with the increase of $\alpha$, the clustering performance first increases then becomes stable between 1 and 1000. The reason is that when $\alpha$ is small, the sparsity of the model is low, which is not



(a) Effect of $\alpha$.

(b) Effect of $\beta$.

Figure 2: Parameter study of NetFS on Epinions.

suitable for feature selection. Then we fix $\alpha$ to be 10 and vary $\beta$ in $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$, the results are presented in Figure 2(b), we can observe that the clustering performance is less sensitive to $\beta$ compared with $\alpha$, the performance is relatively better when $\beta$ is around 1. The clustering performance is relatively more sensitive to the number of selected features, which is still an open problem in unsupervised feature selection.

## 5 Related Work

In this section, we review some related work from two aspects: (1) traditional feature selection; and (2) feature selection for networked data.

**5.1 Traditional Feature Selection** Based on the availability of class labels, feature selection algorithms can be categorized into supervised and unsupervised methods. Supervised feature selection algorithms evaluate feature relevance by its correlation to the class labels. Supervised methods can be further divided into wrapper methods and filter methods [15, 22]. Wrapper methods require a predefined learning algorithm and use the learning performance to assess features. The computational cost of wrappers is usually expensive [11]. Filter methods are independent of any learning algorithms, they rely on some characteristics of data like distance, consistency, dependency and correlation to evaluate the importance of features [9, 12, 26]. Since most real-world data is unlabeled, more and more researchers pay attention to develop effective and efficient unsupervised feature selection algorithms. Due to the lack of label information, they exploit different alternative criteria to assess feature relevance such as data similarity [3, 12, 34], local discriminative information [19, 33] and data reconstruction error [7]. Recently, unsupervised feature selection methods that uses $\ell_1$-norm and $\ell_{2,1}$-norm [3, 19, 33, 27] regularization have been widely investigated and proven to achieve good performance, through the sparsity regularization, feature selection can be embedded in the learning process.

**5.2 Feature Selection for Networked Data** Traditional feature selection algorithms cannot be directly applied on networked data as the *i.i.d.* assumption does not hold. Gu and Han [8] proposed a supervised feature selection method for networked data. They exploit a linear regression model to capture the content information and adopt graph regularization to take into account link information. Tang and Liu [28] first attempted to perform feature selection on social media data. In the proposed supervised feature selection framework, different social relationships (CoPost, CoFollowing, CoFollowed and Following) are extracted to enhance the feature selection performance. Since social media data is ubiquitous and effortless to label, an unsupervised feature selection algorithm for linked social media data, LUFS, is proposed in [29]. Social dimensions are extracted and linked data are exploited to help select relevant features. USFS [17] is an unsupervised streaming feature selection algorithm which takes advantage of external latent representations to update selected feature set timely. However, these works are substantially different from our proposed framework NetFS in either of the following - (1) NetFS is an unsupervised feature selection method which leverages both link and content information, without label information to assess feature importance, the task is more challenging; and (2) NetFS provides an iterative way to learn latent representations and feature weights in a joint learning framework and the feature selection phase is more robust to noisy links.

## 6 Conclusions and Future Work

In this paper, we propose a robust unsupervised feature selection framework NetFS for networked data. First, we propose to use latent representations to capture inherent correlations in the network, then we propose to embed the latent representations learning process into feature selection. Therefore, the latent representation learning on the network structure and feature selection on the content space could help and boost each other to obtain good features, and the proposed model is more robust to noisy links. Methodologically, we propose to use an alternating scheme to optimize the objective function of the proposed NetFS framework. Experimental results on real-world social media datasets demonstrate the effectiveness of the proposed framework when measured against the state-of-the-art unsupervised feature selection methods.

There are many future work directions to investigate. First, in this work, we use social media as a test bed to evaluate the proposed NetFS framework, we also would like to validate the proposed framework on other kinds of networks such as gene networks, citation networks. Second, real-world networks are usually not static but evolve over time such that both the network structure and the content information may change, we would like to study how to perform feature selection on dynamic networks in the future.

## Acknowledgments

The authors wish to thank Harsh Dani who assisted in the proofreading of the manuscript. This material is, in part, supported by National Science Foundation (NSF) under grant number IIS-1217466.

## References

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. In *NIPS*, pages 33–40, 2009.

[2] D. P. Bertsekas. *Nonlinear programming*. Athena scientific, 1999.

[3] D. Cai, C. Zhang, and X. He. Unsupervised feature selection for multi-cluster data. In *KDD*, pages 333–342, 2010.

[4] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[5] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205, 2005.

[6] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

[7] A. K. Farahat, A. Ghodsi, and M. S. Kamel. An efficient greedy method for unsupervised feature selection. In *ICDM*, pages 161–170, 2011.

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

[9] Q. Gu, Z. Li, and J. Han. Generalized fisher score for feature selection. In *UAI*, pages 266–273, 2012.

[10] T. Guo, X. Zhu, J. Pei, and C. Zhang. Snoc: streaming network node classification. In *ICDM*, pages 150–159, 2014.

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

[12] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *NIPS*, pages 507–514, 2005.

[13] Z. He, S. Xie, R. Zdunek, G. Zhou, and A. Cichocki. Symmetric nonnegative matrix factorization: Algorithms and applications to probabilistic clustering. *IEEE Transactions on Neural Networks*, 22(12):2117–2131, 2011.

[14] X. Hu, L. Tang, J. Tang, and H. Liu. Exploiting social relations for sentiment analysis in microblogging. In *WSDM*, pages 537–546, 2013.

[15] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[16] D. Kuang, H. Park, and C. H. Ding. Symmetric nonnegative matrix factorization for graph clustering. In *SDM*, volume 12, pages 106–117, 2012.

[17] J. Li, X. Hu, J. Tang, and H. Liu. Unsupervised streaming feature selection in social media. In *CIKM*, pages 1041–1050, 2015.

[18] J. Li and O. Zaiane. Associative classification with statistically significant positive and negative rules. In *CIKM*, pages 633–642, 2015.

[19] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu. Unsupervised feature selection using nonnegative spectral analysis. In *AAAI*, pages 1026–1032, 2012.

[20] C.-J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[21] H. Liu and H. Motoda. *Computational methods of feature selection*. CRC Press, 2007.

[22] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502, 2005.

[23] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

[24] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[25] F. Nie, H. Huang, X. Cai, and C. H. Ding. Efficient and robust feature selection via joint 2, 1-norms minimization. In *NIPS*, pages 1813–1821, 2010.

[26] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[27] M. Qian and C. Zhai. Robust unsupervised feature selection. In *IJCAI*, pages 1621–1627, 2013.

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

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

[30] L. Tang and H. Liu. Relational learning via latent social dimensions. In *KDD*, pages 817–826, 2009.

[31] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[32] Y. Wang, Y. Hu, S. Kambhampati, and B. Li. Inferring sentiment from web images with joint inference on visual and social cues: A regulated matrix factorization approach. In *ICWSM*, pages 473–482, 2015.

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

[34] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *ICML*, pages 1151–1157, 2007.