Unsupervised Feature Selection for Linked Social Media Data

Jiliang Tang
Computer Science and Engineering
Arizona State University
Tempe, AZ 85281
{Jiliang.Tang}@asu.edu

Huan Liu
Computer Science and Engineering
Arizona State University
Tempe, AZ 85281
{Huan.Liu}@asu.edu

ABSTRACT

The prevalent use of social media produces mountains of unlabeled, high-dimensional data. Feature selection has been shown effective in dealing with high-dimensional data for efficient data mining. Feature selection for unlabeled data remains a challenging task due to the absence of label information by which the feature relevance can be assessed. The unique characteristics of social media data further complicate the already challenging problem of unsupervised feature selection, (e.g., part of social media data is linked, which makes invalid the independent and identically distributed assumption), bringing about new challenges to traditional unsupervised feature selection algorithms. In this paper, we study the differences between social media data and traditional attribute-value data, investigate if the relations revealed in linked data can be used to help select relevant features, and propose a novel unsupervised feature selection framework, LUFS, for linked social media data. We perform experiments with real-world social media datasets to evaluate the effectiveness of the proposed framework and probe the working of its key components.

Categories and Subject Descriptors

1.5.2 [**Pattern Recognition**]: Design Methodology—*Feature evaluation and Selection*

General Terms

Algorithms, Theory

Keywords

Unsupervised Feature Selection, Linked Social Media Data, Pseudo-class Label

1. INTRODUCTION

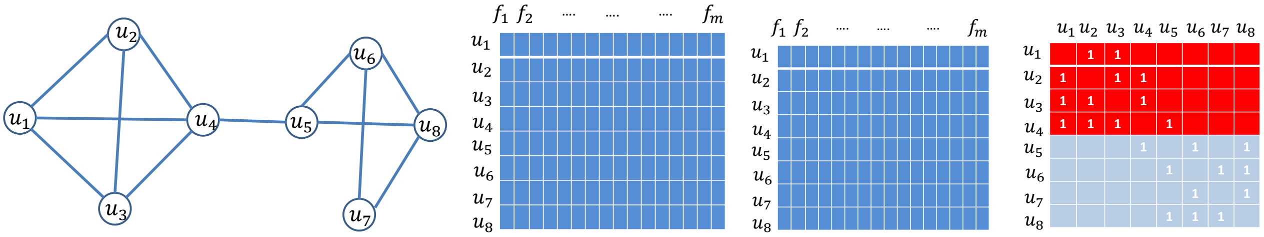
The myriads of social media services such as Facebook and Twitter are available that allow people to communicate and express themselves conveniently and effortlessly. The pervasive use of social media produces massive data at an unprecedented speed. For example, 200 million tweets are posted per day[1]; 3,000 photos are uploaded per minute to Flickr[2]; and the number of Facebook users has increased from 100 million in 2008 to 800 million in 2011[3]. The massive and high-dimensional social media data poses new challenges to data mining tasks such as classification and clustering. One traditional and effective approach to handle high-dimensional data is *feature selection* [7, 18]. Feature selection aims to select relevant features from the high-dimensional data for a compact and accurate data representation. It can alleviate the curse of dimensionality, speed up the learning process, and improve the generalization capability of a learning model [16, 19].

According to whether the training data is labeled or unlabeled, feature selection algorithms can be roughly divided into *supervised* and *unsupervised* feature selection. It is time-consuming and costly to obtain labeled data. Given the scale of social media data, we propose to study unsupervised feature selection. Unsupervised feature selection is particularly difficult due to the absence of class labels for feature relevance assessment [3]. Social media data adds further challenges to feature selection. Most existing feature selection algorithms work with "flat" attribute-value data which is typically assumed to be independent and identically distributed (*i.i.d.*). However, the *i.i.d.* assumption does not hold for social media data since it is inherently linked. Figure 1 shows a simple example of linked data in social media and its two data representations. Figure 1(a) shows 8 linked instances (u_1 to u_8). These instances usually form groups and instances within groups have more connections than instances between groups [31]. For example, u_1 has more connections to $\{u_2, u_3, u_4\}$ than $\{u_5, u_6, u_7, u_8\}$. Figure 1(b) is a conventional representation of attribute-value data: rows are instances and columns are features. For linked data, except for the conventional representation, there is link information between instances as shown in Figure 1(c).

Linked data in social media presents both challenges and opportunities for unsupervised feature selection. In this work, we investigate: (1) *how to exploit and model the relations among data instances*, and (2) *how to take advantage of these relations for feature selection using unlabeled data.* One key difference between supervised and unsupervised fea-

KDD'12, August 12–16, 2012, Beijing, China.
Copyright 2012 ACM 978-1-4503-1462-6 /12/08 ...$10.00.

[1] http://techcrunch.com/2011/06/30/twitter-3200-million-tweets/
[2] http://www.flickr.com/
[3] http://en.wikipedia.org/wiki/Facebook



u_1
u_2
u_3
u_4
u_5
u_6
u_7
u_8
f_1 f_2 .... f_m
u_1 u_2 u_3 u_4 u_5 u_6 u_7 u_8
1

(a) Linked Users (b) Attribute-Value Data (c) Attribute-Value and Linked Data

Figure 1: A Simple Example of Linked Social Media Data

ture selection is the availability of label information. If we change our perspective and consider label information as a sort of constraints in the learning space, we can turn both types of feature selection to a problem with the same intrinsic property: selecting features to be consistent with some constraints [35]. In supervised learning, label information plays the role of constraint. Without labels, other alternative constraints are proposed, such as data variance and separability, for unsupervised learning [27, 7, 4, 14, 35, 3]. However, most of them evaluate the importance of feature individually and neglect the feature correlation [36, 34].

In our attempt to address the challenges of unsupervised feature selection for linked social media data, we propose a novel framework of Linked Unsupervised Feature Selection (LUFS). Our main contributions are summarized below:

- Employing social dimensions to exploit relations among linked data instances in groups and enable a concise mathematical model of these relations;
- Introducing the concept of pseudo-class labels to develop alternative constraints for unsupervised feature selection using linked data;
- Proposing a novel unsupervised feature selection framework, LUFS, for linked data in social media to exploit linked information in selecting features;
- Evaluating LUFS extensively using datasets from real-world social media websites to understanding the working of LUFS.

The rest of this paper is organized as follows. We formally define the problem of unsupervised feature selection for linked data in social media in Section 2; introduce our new framework of unsupervised feature selection, LUFS in Section 3, including social dimension regularization, optimization, and convergence analysis; present empirical evaluation with discussion in Section 4 and the related work in Section 5; and conclude this work in Section 6.

2. PROBLEM STATEMENT

In this paper, scalars are denoted by lower-case letters $(a, b, \ldots; \alpha, \beta, \ldots)$, vectors are written as lower-case bolded letters $(\mathbf{a}, \mathbf{b}, \ldots)$, and matrices correspond to boldfaced upper-case letters $(\mathbf{A}, \mathbf{B}, \ldots)$.

Let $\mathbf{u} = \{u_1, u_2, \ldots, u_n\}$ be the set of linked data (e.g., u_1 to u_8 in Figure 1) where n is the number of data instances and $\mathbf{f} = \{f_1, f_2, \ldots, f_m\}$ be the set of features where m is the number of features. Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) \in \mathbb{R}^{m\times n}$ be the conventional representation of $\mathbf{u}$ w.r.t . $\mathbf{f}$ where $\mathbf{x}_i(j)$ is the frequency of f_j used by u_i (e.g., Figure 1(b)). We assume that the data, $\mathbf{X}$, is centered, that is, $\sum_{i=1}^n \mathbf{x}_i = 0$, which can be realized as $\mathbf{X} = \mathbf{XP}$ where $\mathbf{P} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$.

For social media data, there also exist connections among its data instances. Let $\mathbf{R} \in \mathbb{R}^{n\times n}$ denote their connections where $\mathbf{R}(i, j) = 1$ if u_i and u_j are linked, otherwise zero (e.g., the right subgraph in Figure 1(c)). In our work, we assume that the relationships among data instances form an undirected graph, i.e., $\mathbf{R} = \mathbf{R}^\top$.

With the above notations defined, the problem of *unsupervised feature selection for linked data* can be stated as: given n linked data instances, its conventional representation $\mathbf{X}$, its link representation $\mathbf{R}$, develop a method f, which can select a subset of relevant features $\mathbf{f}'$ from $\mathbf{f}$ by utilizing both $\mathbf{X}$ and $\mathbf{R}$ for these n linked instances, formally stated as,

$$f : \{\mathbf{f}; \mathbf{X}, \mathbf{R}\} \rightarrow \{\mathbf{f}'\}. \qquad (1)$$

The above is substantially different from the traditional task of unsupervised feature selection, formally defined as,

$$f : \{\mathbf{f}; \mathbf{X}\} \rightarrow \{\mathbf{f}'\}. \qquad (2)$$

3. UNSUPERVISED FEATURE SELECTION

Set $\mathbf{s} = \pi(\overbrace{0, \ldots, 0}^{m-k}, \overbrace{1, \ldots, 1}^{k})$ where $\pi(\cdot)$ is the permutation function and k is the number of features to select where $\mathbf{s}_i = 1$ indicates that the i-th feature is selected. The original data can be represented as $\mathrm{diag}(\mathbf{s})\mathbf{X}$ with k selected features, where $\mathrm{diag}(\mathbf{s})$ is a diagonal matrix. The difficulty faced with unsupervised feature selection is due to lack of class labels. Hence, we introduce the concept of pseudo-class label to guide unsupervised learning. We assume that there is a mapping matrix $\mathbf{W} \in \mathbb{R}^{m\times c}$, which assigns each data point with a pseudo-class label where c is the number of pseudo-class labels. The pseudo-class label indicator matrix is $\mathbf{Y} = \mathbf{W}^\top\mathrm{diag}(\mathbf{s})\mathbf{X} \in \mathbb{R}^{c\times n}$. Each column of $\mathbf{Y}$ has only one nonzero entity, i.e., $\|\mathbf{Y}(:, i)\|_0 = 1$ where $\|\cdot\|_0$ is the vector zero norm, counting the number of nonzero elements in the vector. Since $\mathbf{X}$ is centered, it is easy to verify that $\mathbf{Y}$ is also centered,

$$\sum_{i=1}^n \mathbf{y}_i = \sum_{i=1}^n \left(\mathbf{W}^\top\mathrm{diag}(\mathbf{s})\mathbf{x}_i\right) = \left(\mathbf{W}^\top\mathrm{diag}(\mathbf{s})\right)\sum_{i=1}^n \mathbf{x}_i = 0. \qquad (3)$$

As shown above, both supervised and unsupervised learning tasks aim to solve the same problem: selecting features consistent with given constraints. In the supervised setting, pseudo-class label information is tantamount to the provided label information; in the unsupervised setting, we seek pseudo-class label information from linked social media data. We discuss technical details of the proposed framework LUFS in the following subsection.

3.1 Social Dimensions for Linked Data

First, we embark on dealing with linked data. In [30], social dimension is introduced as a means to integrate the interdependency among linked data and attribute-value data. Instances from different social dimensions are dissimilar while instances in the same social dimension are similar. Flattening linked data with social dimensions, traditional classification methods such as SVM obtain better performance for relational learning [30].

Social dimension extraction is a well-studied problem by social network analysis community [24, 11, 30]. In our work, we adopt a widely used algorithm, i.e., Modularity Maximization [24], which can formulated as:

$$\max_{\mathbf{H}^\top\mathbf{H}=\mathbf{I}} \quad Tr\big(\mathbf{H}^\top\mathbf{M}\mathbf{H}\big) \tag{4}$$

where $\mathbf{H} \in \mathbb{R}^{K\times n}$ is the social dimension indicator matrix, K is the number of social dimensions, and $\mathbf{M}$ is the modularity matrix defined as:

$$\mathbf{M} = \mathbf{R} - \frac{\mathbf{d}\mathbf{d}^\top}{2n} \tag{5}$$

where $\mathbf{d}$ is a degree vector, and $\mathbf{d}_i$ is the degree of u_i.

The standard K-means algorithm is performed to obtain the discrete-valued social dimension assignment, $\mathbf{H}(i,j) = 1$, if u_j is in the i-th dimension, and $\mathbf{H}(i,j) = 0$, otherwise.

Recall the simple example of linked data in Figure 1, we can extract two social dimensions, i.e, $\{u_1, u_2, u_3, u_4\}$ and $\{u_5, u_6, u_7, u_8\}$. From Figure 1, we can observe that instances in the same social dimension have more connections. The discrete-valued social dimension indicator matrix, $\mathbf{H}$, is:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Since social dimensions represent a type of user affiliations, inspired by Linear Discriminant Analysis, we define three matrices: within, between, and total social dimension scatter matrixes $\mathbf{S}_w$, $\mathbf{S}_b$, and $\mathbf{S}_t$ as follows:

$$\begin{aligned} \mathbf{S}_w &= \mathbf{Y}\mathbf{Y}^\top - \mathbf{Y}\mathbf{F}\mathbf{F}^\top\mathbf{Y}^\top, \\ \mathbf{S}_b &= \mathbf{Y}\mathbf{F}\mathbf{F}^\top\mathbf{Y}^\top, \\ \mathbf{S}_t &= \mathbf{Y}\mathbf{Y}^\top, \end{aligned} \tag{6}$$

where $\mathbf{F}$ is the weighted social dimension indicator matrix, which can be obtained from $\mathbf{H}$ as below,

$$\mathbf{F} = \mathbf{H}(\mathbf{H}^\top\mathbf{H})^{-\frac{1}{2}}, \tag{7}$$

where the j-th column of $\mathbf{F}$ is given by

$$F_j = \frac{(0, \ldots, 0, \overbrace{1, \ldots, 1}^{h_j}, 0, \ldots, 0)}{\sqrt{h_j}}. \tag{8}$$

where h_j is the number of instances in the j-th social dimension.

Instances in the same social dimension are similar and instances from different social dimensions are dissimilar. We can obtain a constraint from link information, **social dimension regularization**, to model the relations among linked instances, via the following maximization problem,

$$\max_{\mathbf{W}} Tr\big((\mathbf{S}_t)^{-1}\mathbf{S}_b\big). \tag{9}$$

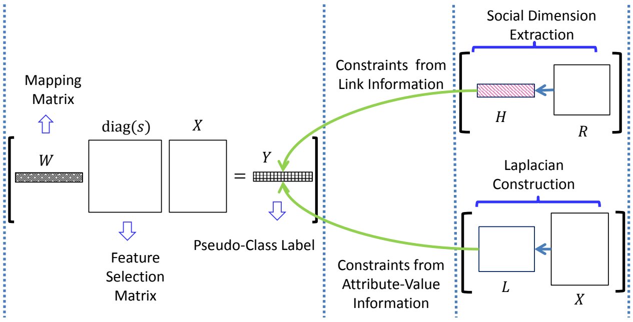

Mapping Matrix
diag(s)
X
W
=
Y
Feature Selection Matrix
Pseudo-Class Label
Constraints from Link Information
Social Dimension Extraction
H
R
Constraints from Attribute-Value Information
Laplacian Construction
L
X

Figure 2: Illustration of the LUFS Framework

3.2 A Framework - LUFS

To take advantage of information from attribute-value part, i.e., $\mathbf{X}$, similar data instances should have similar labels. According to spectral analysis [22], the constraint from attribute-value part can be formulated as the following minimization problem,

$$\min \quad Tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^\top) \tag{10}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is a laplacian matrix and $\mathbf{D}$ is a diagonal matrix with its elements defined as $\mathbf{D}(i,i) = \sum_{j=1}^{n} \mathbf{S}(i,j)$. $\mathbf{S} \in \mathbb{R}^{n\times n}$ denotes the similarity matrix based on $\mathbf{X}$, obtained through a RBF kernel as,

$$\mathbf{S}(i,j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}}. \tag{11}$$

By introducing the concept of *pseudo-class labels*, constraints from both link information and unlabeled attribute-value data are ready for unsupervised feature selection. Our framework of linked unsupervised feature selection, LUFS, is conceptually shown in Figure 2 with our solutions to the two challenges (need to take into account of linked data and lack of labels): extracting constraints from both linked and attribute-value data, and then constructing pseudo-class labels through social dimension extraction and spectral analysis. Thus, our proposed framework, LUFS, is equivalent to solving the following optimization problem,

$$\begin{aligned} \min_{\mathbf{W},\mathbf{s}} \quad & Tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^\top) - \alpha Tr\big((\mathbf{S}_t)^{-1}\mathbf{S}_b\big), \\ s.t. \quad & \mathbf{s} \in \{0,1\}^n, \ \mathbf{s}^\top\mathbf{1}_n = k, \\ & \|\mathbf{Y}(:,i)\|_0 = 1, \quad 1 \le i \le n. \end{aligned} \tag{12}$$

The problem in Eq. (12) is difficult to solve due to the two constraints. To make the problem solvable, we use some widely used relaxation methods. First, according to common relaxation for label indicator matrix [22], the constraint on $\mathbf{Y}$ is relaxed to orthogonality, i.e., $\mathbf{Y}\mathbf{Y}^\top = \mathbf{I}_c$.

With this relaxation, $\mathbf{S}_t$ in social dimension regularization is an identity matrix and $Tr(\mathbf{Y}\mathbf{Y}^\top) = c$ is a constant. Social dimension regularization can be rewritten as,

$$\min Tr(\mathbf{Y}\mathbf{Y}^\top) - Tr(\mathbf{Y}\mathbf{F}\mathbf{F}^\top\mathbf{Y}^\top). \tag{13}$$

the reason for adding the constant in social dimension regularization will be explained later. With this relaxation, Eq. (12) can be relaxed to,

$$\begin{aligned} \min & Tr(\mathbf{YLY}^\top) + \alpha Tr(\mathbf{YY}^\top - \mathbf{YFF}^\top\mathbf{Y}^\top) \\ s.t. & \quad \mathbf{s} \in \{0,1\}^n,\ \mathbf{s}^\top\mathbf{1}_n = k, \\ & \quad \mathbf{YY}^\top = \mathbf{I}_c, \end{aligned} \tag{14}$$

The constraint on $\mathbf{s}$ makes Eq. (14) mixed integer programming [2], which is still difficult to solve. We observe that that diag($\mathbf{s}$) and $\mathbf{W}$ is as the form of diag($\mathbf{s}$)$\mathbf{W}$ in Eq. (14). Since $\mathbf{s}$ is a binary vector and $m - k$ rows of the diag($\mathbf{s}$) are all zeros, diag($\mathbf{s}$)$\mathbf{W}$ is a matrix where the elements of many rows are all zeros. This motivates us to absorb the diag($\mathbf{s}$) into $\mathbf{W}$, $\mathbf{W}$ = diag($\mathbf{s}$)$\mathbf{W}$, and add $\ell_{2,1}$ norm on $\mathbf{W}$ to achieve feature selection as,

$$\begin{aligned} \min_{\mathbf{W}} & \quad Tr(\mathbf{W}^\top\mathbf{XLX}^\top\mathbf{W}) + \beta\|\mathbf{W}\|_{2,1} \\ & + \alpha Tr\big(\mathbf{W}^\top\mathbf{X}(\mathbf{I}_n - \mathbf{FF}^\top)\mathbf{X}^\top\mathbf{W}\big) \\ s.t. & \quad \mathbf{W}^\top(\mathbf{XX}^\top + \lambda\mathbf{I})\mathbf{W} = \mathbf{I}_c, \end{aligned} \tag{15}$$

where $\lambda\mathbf{I}$ is added to make $(\mathbf{XX}^\top + \lambda\mathbf{I})$ nonsingular. $\|\mathbf{W}\|_{2,1}$, controls the capacity of $\|\mathbf{W}\|$ and also ensures that $\|\mathbf{W}\|$ is sparse in rows, making it particularly suitable for feature selection. $\|\mathbf{W}\|_{2,1}$ is the $\ell_{2,1}$-norm of $\|\mathbf{W}\|$ defined as [5],

$$\|\mathbf{W}\|_{2,1} = \sum_{i=1}^{m}\sqrt{\sum_{j=1}^{k}\mathbf{W}^2(i,j)} = \sum_{i=1}^{m}\|\mathbf{W}(i,:)\|_2. \tag{16}$$

Lemma 1. Assume $\mathbf{E} \in \mathbb{R}^{n\times m}$ is a matrix with orthonormal columns, that is, $\mathbf{E}^\top\mathbf{E} = \mathbf{I}_m$, then $\mathbf{I}_n - \mathbf{EE}^\top$ is a symmetric and positive semidefinite matrix.

PROOF. It is easy to verify that $\mathbf{I}_n - \mathbf{EE}^\top$ is a symmetric matrix since $(\mathbf{I}_n - \mathbf{EE}^\top)^\top = \mathbf{I}_n - \mathbf{EE}^\top$.

Let $\mathbf{E} = \mathbf{U}\Sigma\mathbf{V}^\top$ be the Singular Value Decomposition (SVD) of $\mathbf{E}$, where $\mathbf{U} \in \mathbb{R}^{n\times n}$ and $\mathbf{V} \in \mathbb{R}^{m\times m}$ are orthogonal. $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_m, 0, \ldots, 0)$ is diagonal where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_m \geq 0$. Then we have,

$$\begin{aligned} & \mathbf{E}^\top\mathbf{E} = \mathbf{I}_m \Rightarrow \mathbf{V}\Sigma^\top\Sigma\mathbf{V}^\top = \mathbf{VV}^\top \\ & \Rightarrow 1 \geq \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_m \geq 0. \end{aligned} \tag{17}$$

Then,

$$\mathbf{I}_n - \mathbf{EE}^\top = \mathbf{U}(\mathbf{I}_n - \Sigma\Sigma^\top)\mathbf{U}^\top. \tag{18}$$

Since $(\mathbf{I}_n - \Sigma\Sigma^\top)$ is a diagonal matrix and the diagonal elements are all nonnegative, $(\mathbf{I}_n - \Sigma\Sigma^\top)$ is a positive semidefinite matrix. Thus $\mathbf{U}(\mathbf{I}_n - \Sigma\Sigma^\top)\mathbf{U}^\top$ is a semidefinite matrix, which completes the proof. □

The importance of Lemma 1 is two-fold. First, it explains why we need a constant on the social dimension regularization. Adding the constant can guarantee the convexity of the objective function in Eq. (15). Second, it paves the way for the following theorem for LUFS.

THEOREM 1. *Eq. (15) can be converted into the following optimization problem,*

$$\begin{aligned} \min_{\mathbf{W}} & \quad f(\mathbf{W}) = Tr(\mathbf{W}^\top\mathbf{AW}) + \beta\|\mathbf{W}\|_{2,1}, \\ s.t. & \quad \mathbf{W}^\top\mathbf{BW} = \mathbf{I}_c, \end{aligned} \tag{19}$$

where $\mathbf{A}$ *is a symmetric and positive semidefinite matrix and* $\mathbf{B}$ *is a symmetric and positive matrix.*

PROOF. It suffices to show how to construct a symmetric and positive semidefinite matrix $\mathbf{A}$ and a symmetric and positive matrix $\mathbf{B}$ from Eq. (15).

We construct $\mathbf{B} = \mathbf{XX}^\top + \lambda\mathbf{I}$. It is easy to check that $\mathbf{B}$ is symmetric and positive when $\lambda \neq 0$ and then the constraint in Eq. (15) is converted into the one in Eq. (19).

We construct $\mathbf{A} = \mathbf{XLX}^\top + \alpha\mathbf{X}(\mathbf{I}_n - \mathbf{FF}^\top)\mathbf{X}^\top$ and then the objective function in Eq. (15) is converted into the one in Eq. (19). According to Lemma 1, $(\mathbf{I}_n - \mathbf{FF}^\top)$ is symmetric and positive semidefinite; and according to the definition of Laplacian matrix, $\mathbf{XLX}^\top$ is symmetric and positive semidefinite. Thus, $\mathbf{A}$ is symmetric and positive semidefinite, which completes the proof. □

3.3 Optimization Algorithm for LUFS

In recent years, many methods have been proposed to solve the $\ell_{2,1}$-norm minimization problem [21, 25, 34]. However, our problem is different from these existing ones due to the orthogonal constraint in Eq. (19). Hence, we propose the following algorithm, as shown in Algorithm 1, to optimize the problem in Eq. (19).

Algorithm 1 LUFS

```
Input: $\{\mathbf{X}, \mathbf{R}, \alpha, \beta, \lambda, c, K, k\}$
Output: $k$ most relevant features
 1: Obtain the social dimension indicator matrix $\mathbf{H}$
 2: Set $\mathbf{F} = \mathbf{H}(\mathbf{H}^\top\mathbf{H})^{-\frac{1}{2}}$
 3: Construct $\mathbf{S}$ through Eq. (11)
 4: Set $\mathbf{L} = \mathbf{D} - \mathbf{S}$
 5: Set $\mathbf{A} = \mathbf{XLX}^\top + \alpha\mathbf{X}(\mathbf{I}_n - \mathbf{FF}^\top)\mathbf{X}^\top$
 6: Set $\mathbf{B} = \mathbf{XX}^\top + \lambda\mathbf{I}$
 7: Set $t = 0$ and initialize $\mathbf{D}_0$ as an identity matrix
 8: while Not convergent do
 9:     Set $\mathbf{C}_t = \mathbf{B}^{-1}(\mathbf{A} + \beta\mathbf{D}_t)$
10:     Set $\mathbf{W}_t = [q_1, \ldots, q_c]$ where $q_1, \ldots, q_c$ are the eigenvectors of $\mathbf{C}_t$ corresponding to the first $c$ smallest eigenvalues
11:     Update the diagonal matrix $\mathbf{D}_{t+1}$, where the $i$-th diagonal element is $\frac{1}{2\|\mathbf{W}_t(i,:)\|_2}$;
12:     Set $t = t + 1$
13: end while
14: Sort each feature according to $\|\mathbf{W}(i,:)\|_2$ in descending order and select the top-$k$ ranked ones;
```

In Algorithm 1, social dimension extraction and weighted social dimension indicator construction are from line 1 to line 2. The iterative algorithm to optimize Eq. (19) is presented from line 8 to line 13. The convergence analysis of the algorithm starts with the following two lemmas.

Lemma 2. $\mathbf{A} \in \mathbb{R}^{m\times m}$ is symmetric and positive semidefinite and $\mathbf{B} \in \mathbb{R}^{m\times m}$ is symmetric and positive. If $\mathbf{W} \in \mathbb{R}^{m\times c}$ solves the following minimization problem:

$$\begin{aligned} \min_{\mathbf{W}} & \quad Tr(\mathbf{W}^\top\mathbf{AW}), \\ s.t. & \quad \mathbf{W}^\top\mathbf{BW} = \mathbf{I}_c, \end{aligned} \tag{20}$$

then $\mathbf{W}$ consists of the eigenvectors of $\mathbf{B}^{-1}\mathbf{A}$ corresponding to the c smallest eigenvalues.

PROOF. The lemma can be easily obtained through *Ky Fan Theorem* [15]. □

Lemma 3. The following inequality holds if $\mathbf{w}_t^i|_{i=1}^r$ are non-zero vectors, where r is an arbitrary number.

$$\sum_i \|\mathbf{w}_{t+1}^i\|_2 - \sum_i \frac{\|\mathbf{w}_{t+1}^i\|_2}{2\|\mathbf{w}_t^i\|_2} \leq \sum_i \|\mathbf{w}_t^i\|_2 - \sum_i \frac{\|\mathbf{w}_t^i\|_2^2}{2\|\mathbf{w}_t^i\|_2}. \quad (21)$$

Proof. The detailed proof can be found in our previous work [29]. □

With the above two lemmas, we develop the following theorem regarding the convergence of Algorithm 1.

Theorem 2. *At each iteration of Algorithm 1, the value of the objective function in Eq. (19) monotonically decreases.*

Proof. According to Lemma 2, $\mathbf{W}_{t+1}$ in line 9 of Algorithm 1 is the solution to the following problem,

$$\mathbf{W}_{t+1} = \min_{\mathbf{W}^\top \mathbf{B}\mathbf{W}=\mathbf{I}} Tr\big(\mathbf{W}^\top(\mathbf{A}+\beta\mathbf{D}_t)\mathbf{W}\big), \quad (22)$$

which indicates that,

$$Tr\big(\mathbf{W}_{t+1}^\top(\mathbf{A}+\beta\mathbf{D}_t)\mathbf{W}_{t+1}\big) \leq Tr\big(\mathbf{W}_t^\top(\mathbf{A}+\beta\mathbf{D}_t)\mathbf{W}_t\big).$$

Then we have the following inequality,

$$\begin{aligned} & Tr(\mathbf{W}_{t+1}^\top\mathbf{A}\mathbf{W}_{t+1}) + \beta\sum_i \|\mathbf{W}_{t+1}(i,:)\|_2 \\ & - \beta(\sum_i \|\mathbf{W}_{t+1}(i,:)\|_2 - \sum_i \frac{\|\mathbf{W}_{t+1}(i,:)\|_2^2}{2\|\mathbf{W}_t(i,:)\|_2}) \\ \leq\ & Tr(\mathbf{W}_t^\top\mathbf{A}\mathbf{W}_t) + \beta\sum_i \|\mathbf{W}_t(i,:)\|_2 \\ & - \beta(\sum_i \|\mathbf{W}_t(i,:)\|_2 - \sum_i \frac{\|\mathbf{W}_t(i,:)\|_2^2}{2\|\mathbf{W}_t^(i,:)\|_2}) \end{aligned}$$

According to the lemma 3, we can obtain,

$$f(\mathbf{W}_{t+1}) \leq f(\mathbf{W}_t), \quad (23)$$

which completes the proof. □

According to Theorem 2, Algorithm 1 converges to the optimal $\mathbf{W}$ for the problem in Eq. (19).

4. EXPERIMENTS AND DISCUSSION

In this section, we present experiment details to verify the effectiveness of the proposed framework, LUFS. After introducing real-world social media datasets, we first evaluate the quality of selected features in terms of clustering performance, then study the effects of parameters on performance and finally further verify the constraint extracted from link information by social dimension.

4.1 Datasets, Baseline Methods, and Metrics

We collect two datasets from real-world social media websites, i.e., BlogCatalog[4] and Flickr[5], which are the subsets of two public available datasets used in [31] to uncover overlapping groups in social media. Some statistics of the datasets are shown in Table 1.

Table 1: Statistics of the Datasets

	BlogCatalog	Flickr
Size	5,198	7,575
# of Features	8,189	12,047
# of Classes	6	9
# of Links	27,965	47,344
# Ave Degree	5.38	6.25

LUFS is compared with the following three unsupervised feature selection algorithms,

- UDFS [34] selects features in batch mode by simultaneously exploiting discriminative information and feature correlation.
- Laplacian Score [14] evaluates the importance of a feature through its power of locality preservation.
- SPEC [35] selects features using spectral regression.

Following the existing evaluation practice for unsupervised feature selection, we assess LUFS in terms of clustering performance. We vary the numbers of selected features as $\{200, 300, 400, 500, 600, 700, 800, 900, 1000\}$. Each feature selection algorithm is first performed to select features, then K-means clustering is performed based on the selected features. Since K-means often converges to local minima, we repeat each experiment 20 times and report the average performance.

The clustering quality is evaluated by two commonly used metrics, *accuracy* and *normalized mutual information* (NMI)[6]. Denoting $l(c_i)$ as the label of cluster c_i, $l(d_j)$ as the predicted label of the j-th document, the accuracy is defined as:

$$Accuracy = \frac{1}{n}\sum_{i=1}^{K}\sum_{j=1}^{n}\delta(l(c_i), l(d_j)), \quad (24)$$

where $\delta(x,y)$ is the delta function that its value is 1 if $x=y$ and 0 otherwise.

Given two clusterings C and C', the mutual information $MI(C,C')$ is defined as:

$$MI(C,C') = \sum_{c_i\in C, c_j'\in C'} p(c_i, c_j')\log_2\frac{p(c_i,c_j')}{p(c_i)p(c_j')}, \quad (25)$$

and NMI is defined by

$$NMI(C,C') = \frac{MI(C,C')}{\max(H(C),H(C'))}, \quad (26)$$

where $H(C)$ and $H(C')$ represent the entropies of clusterings C and C', respectively. Larger NMI values represent better clustering qualities.

4.2 Quality of Selected Features

In this subsection, we compare the quality of features selected by different algorithms using performance metrics given above. Conventionally, the parameters in feature selection algorithms are tuned via cross-validation. More discussion is given in Section 4.3. The resulting parameter values for LUFS are: $\{\alpha = 0.1, \beta = 0.1, K = 70, c = 9\}$ for Flickr while $\{\alpha = 0.1, \beta = 0.1, K = 10, c = 6\}$ for BlogCatalog. λ is used to make $(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})$ nonsingular and according to empirical experience, we set λ to 0.01 in both

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

[5] http://www.flickr.com/

[6] We use the source code from http://www.zjucadcg.cn/dengcai/Data/Clustering.html.

datasets. The comparison results of Flickr and BlogCatalog are shown in Tables 2 and 3, respectively. Note that the clustering performance with all features (i.e., without feature selection) is reported in the last columns.

We observe the performance change with the numbers of selected features: it increases, reaches the peak, and then decreases. For example, LUFS achieves its peak values when the number of selected features are 500 and 300 for Flickr and BlogCatalog, respectively. The clustering performance with as few as 200 features is better than that with all features. For instances, LUFS obtains 10.51% and 18.68% relative improvement in terms of accuracy for Flickr and BlogCatalog, respectively. These results demonstrate that the number of features can be significantly reduced without performance deterioration.

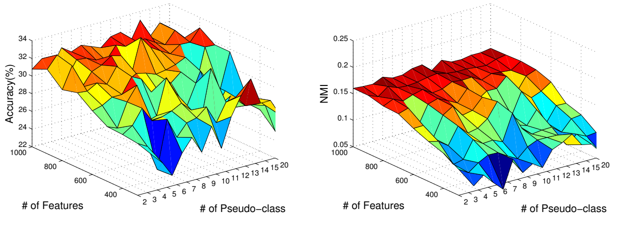
LapScore obtains comparable results with SPEC on both datasets. Most of time, UDFS outperforms both LapScore and SPEC, which is consistent with the results reported in [34]. LapScore and SPEC analyze features separately and select features one after another, which may omit the possible correlation between different features. While UDFS selects features in a batch mode and considers feature correlation. These observations support the conclusion in [3, 36, 34]: it is recommended to analyze data features jointly for feature selection.

We observe that LUFS consistently outperforms these baseline methods on both datasets. For example, LUFS gains up to 19.82% and 11.44% relative improvement in terms of accuracy in Flickr and BlogCatalog, respectively. All baseline methods are based on the data *i.i.d.* assumption, which is not valid due to linked data in social media; and LUFS explicitly takes advantage of link information through social dimension regularization. We also note that with the increasing number of features, the improvement progressively decreases. Usually, LUFS achieves its best results sooner than baseline methods do. For example, LUFS achieves its best results in terms of accuracy in Flickr when 500 features are selected compared to 900 features for baseline methods.

4.3 Parameter Selection

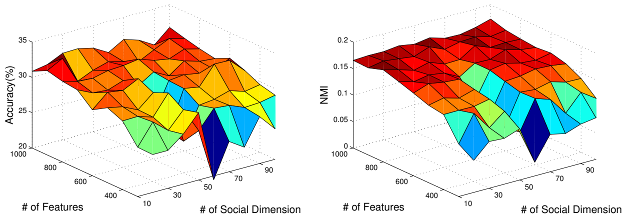
In addition to determining the number of selected features (which remains an open problem [34]), LUFS has four important parameters: the number of pseudo-class labels (c), the number of social dimensions (K), α (controlling social dimension regularization) and β (controlling $\ell_{2,1}$-norm regularization). Hence, we study the effect of each of the four parameters (c, K, α, or β) by fixing the other 3 to see how the performance of LUFS varies with the number of selected features. The processes of parameter selection for Flickr and for BlogCatalog are similar and we present the details for Flickr to save space. Examples of experimental results are presented next.

The number of pseudo-class labels c is varied from 2 to 20 with an incremental step of 1 while setting $\{\alpha = 0.1, \beta = 0.1, K = 70\}$. The performance variation w.r.t. c and the number of features is depicted in Figure 3. Most of the time, the clustering performance first increases rapidly, reaches its best performance and decreases as c increases. This observation can be used to guide the determination of the optimal number of c for LUFS. Note that when c varies from 5 to 11, the clustering performance is not sensitive to c, especially when the number of selected feature is large.



(a) Accuracy (b) NMI

Figure 3: Number of Pseudo-class Labels vs Number of Features



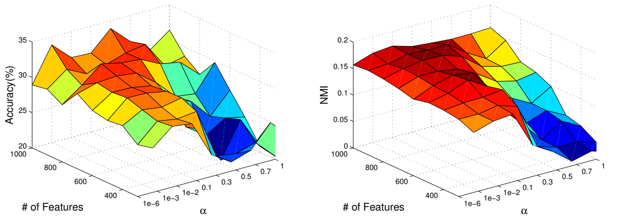
(a) Accuracy (b) NMI

Figure 4: Number of Social Dimension vs Number of Features

Fixing $c = 9$, $\alpha = 0.1$ and $\beta = 0.1$, we vary the number of social dimensions from 10 to 100 with an incremental step of 10 and the performance variation w.r.t. the number of social dimensions and the number of features is demonstrated in Figure 4. Most of the time, with the increasing number of social dimensions, the performance first increases, reaches its peak value and degrades. This pattern can be used to determine the optimal number of social dimensions for LUFS.

Fixing $c = 9$, $K = 70$ and $\beta = 0.1$, we vary α as $\{1e{-}6, 1e{-}3, 1e{-}2, 0.1, 0.3, 0.5, 0.7, 1\}$. The performance variation w.r.t. α and the number of features is depicted in Figure 5. The performance first increases and most of time, the peak values for both accuracy and NMI are achieved when $\alpha = 0.1$, indicating the importance of social dimension regularization for LUFS. After $\alpha = 0.5$, the performance is dramatically degraded, suggesting that only link information is not enough for LUFS.

To study how β and the number of features affect the performance, we vary β as $\{1e{-}6, 1e{-}3, 1e{-}2, 0.1, 0.3, 0.5, 0.7, 1\}$ and set $c = 9$, $K = 70$ and $\beta = 0.1$. The results are shown in Figure 6. We observe that the performance improves as β changes from $1e{-}3$ to $1e{-}2$ and from $1e{-}2$ to 0.1. These results demonstrate the capability of the $\ell_{2,1}$-norm for feature selection.



(a) Accuracy (b) NMI

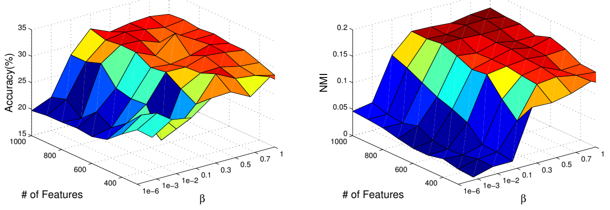
Figure 5: α vs Number of Features

Table 2: Clustering Performance with Different Feature Selection Algorithms in Flickr

	Accuracy									
	200	300	400	500	600	700	800	900	1000	12047
UDFS	26.29	26.41	27.02	27.29	27.55	27.95	28.41	31.07	30.79	25.49
LapScore	25.56	25.79	25.98	26.00	26.00	26.00	26.54	30.90	30.73	25.49
SPEC	26.29	26.29	26.29	26.29	26.29	26.29	26.29	26.35	25.62	25.49
LUFS	29.19	29.51	29.90	32.70	30.41	31.17	30.48	31.79	31.17	25.49
	NMI									
	200	300	400	500	600	700	800	900	1000	12047
UDFS	0.0361	0.0409	0.0702	0.0761	0.0811	0.0921	0.1062	0.1110	0.1200	0.0296
LapScore	0.0302	0.0555	0.0691	0.0709	0.0662	0.0630	0.0694	0.0816	0.1056	0.0296
SPEC	0.0361	0.0361	0.0361	0.0361	0.0361	0.0361	0.0361	0.0361	0.0559	0.0296
LUFS	0.0951	0.1026	0.1489	0.1601	0.1582	0.1701	0.1614	0.1681	0.1596	0.0296

Table 3: Clustering Performance with Different Feature Selection Algorithms in BlogCatalog

	Accuracy									
	200	300	400	500	600	700	800	900	1000	8189
UDFS	41.16	51.86	49.28	49.01	47.96	47.31	46.98	46.98	45.36	38.65
LapScore	41.07	49.83	48.93	48.59	47.75	46.84	46.69	46.76	45.09	38.65
SPEC	40.33	51.16	48.55	48.66	47.43	46.93	46.73	46.60	45.54	38.65
LUFS	45.87	55.70	52.72	51.41	50.93	50.68	50.35	48.95	47.20	38.65
	NMI									
	200	300	400	500	600	700	800	900	1000	8189
UDFS	0.1842	0.1807	0.1789	0.1764	0.1741	0.1725	0.1744	0.1702	0.1673	0.1540
LapScore	0.1328	0.1755	0.1707	0.1746	0.1753	0.1757	0.1730	0.1675	0.1615	0.1540
SPEC	0.1819	0.1789	0.1768	0.1726	0.1765	0.1688	0.1742	0.1675	0.1654	0.1540
LUFS	0.2149	0.2121	0.2026	0.2040	0.2016	0.2014	0.1994	0.1927	0.1891	0.1540



(a) Accuracy (b) NMI

Figure 6: β vs Number of Features

Among the these four parameters of LUFS, β is most sensitive, the number of pseudo-class labels, the number of social dimensions and α are not so.

4.4 Probing Further

A key contribution of LUFS to the performance improvement is to employ social dimensions extracted from linked data. Hence, we would like to probe further why the use of social dimensions works. One way is to investigate whether instances in the same social dimension are similar and instances from different social dimensions are dissimilar.

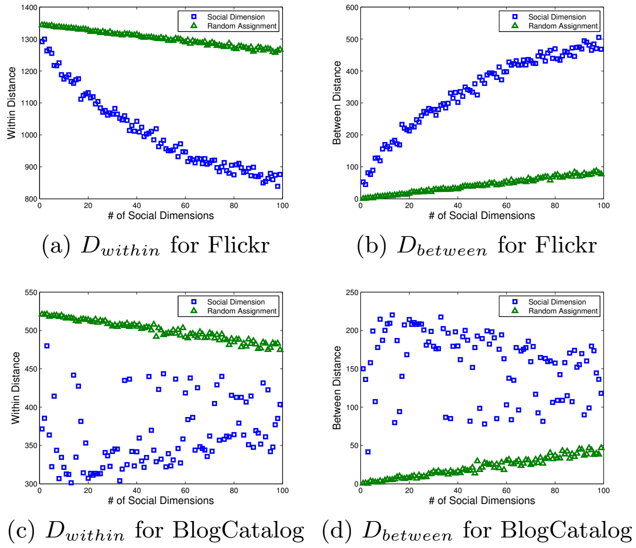
Let $\mathbf{z} = \{z_1, z_2, \ldots, z_K\}$ be the K social dimensions given by the social dimension extraction algorithm, i.e., Modularity Maximization [24] here, with sizes of $\{n_1, n_2, \ldots, n_K\}$ where n_i is number of instances in the i-th social dimension and $\sum_i n_i = n$. To create reference groups in comparison with social dimensions, we also randomly divide these n instances into K groups with sizes of $\{n_1, n_2, \ldots, n_K\}$. Let $\mathbf{z}' = \{z'_1, z'_2, \ldots, z'_K\}$ be the set of these randomly formed groups with sizes of $\{n_1, n_2, \ldots, n_K\}$ and then each group in $\mathbf{z}'$ (e.g., z'_i) corresponds to a social dimension in $\mathbf{z}$ (e.g., z_i). The label information is used earlier in assessing clustering performance. We use it again here. Let $\mathbf{Y}'$ be the class label indicator matrix. We center $\mathbf{Y}'$ as: $\mathbf{Y}' = \mathbf{Y}'\mathbf{P}$. Two distance metrics are defined: D_{within} and $D_{between}$ for within- and between-social dimension distance. D_{within} and $D_{between}$ can be obtained from within social dimension scatter matrix $\mathbf{S}_w$ and between social dimension scatter matrix $\mathbf{S}_b$, respectively,

$$D_{within} = Tr(\mathbf{S}_w), \quad D_{between} = Tr(\mathbf{S}_b). \tag{27}$$

For each specific number of social dimensions, K, we calculate D_{within} and $D_{between}$ for $\mathbf{z}$ and $\mathbf{z}'$. Varying K from 2 to 100 with an incremental step of 1, we obtain 99 pairs of D_{within} and 99 pairs of $D_{between}$. The results for Flickr and BlogCatalog are shown in Figure 7.

D_{within} and $D_{between}$ change much faster for social dimensions, comparing with groups of random assignment. Moreover, D_{within} of a social dimension is much smaller than that of a random assignment group, thus, instances in the same social dimension are of similar labels. $D_{between}$ of a social dimension is much larger than that of a random assignment group, indicating that instances from different social dimensions are dissimilar.

We also perform a two-sample t-test on these pairs of D_{within} of $\mathbf{z}$ and $\mathbf{z}'$ at significant level 0.001. The null hypothesis, H_0, is that there is no different between these pairs; the alternative hypothesis, H_1, is that D_{within} of a social dimension is less than that of the corresponding random assignment group. The t-test results, p-values, are $6.0397e{-}060$ and $8.2406e{-}083$ on Flicker and BlogCatalog,



(a) D_{within} for Flickr (b) $D_{between}$ for Flickr

(c) D_{within} for BlogCatalog (d) $D_{between}$ for BlogCatalog

Figure 7: Within and Between Distance Achieved by Social Dimension and Random Assignment

respectively. Hence, there is strong evidence to reject the null hypothesis. We conduct a similar test for the pairs of $D_{between}$, and there is strong evidence to support that $D_{between}$ of a social dimension is significantly larger than that of its counterpart, a random assignment group. The evidence from both figures and t-test confirms the positive impact of the constraint from link information via social dimensions.

5. RELATED WORK

Traditionally, feature selection algorithms can be either supervised or unsupervised [7, 20] based on the training data being labeled or unlabeled.

Supervised feature selection methods [20] can be broadly categorized into the *wrapper* models [8, 17] and the *filter* models [13, 26]. The wrapper model uses the predictive accuracy of a predetermined learning algorithm to determine the quality of selected features. These methods can be egregiously expensive to run for data with a large number of features [6, 12]. The filter model separates feature selection from classifier learning so that the bias of a learning algorithm does not interact with the bias of a feature selection algorithm. It relies on measures of the general characteristics of the training data such as distance, consistency, dependency, information, and correlation [13]. Many researchers paid great attention to developing unsupervised feature selection [32, 14, 4]. Unsupervised feature selection [14, 7, 35] is a less constrained search problem without class labels, depending on clustering quality measures [10, 9], and can eventuate many equally valid feature subsets. With high-dimensional data, it is likely to find many sets of features that seem equally good without considering additional constraints. Another key difficulty is how to objectively measure the results of feature selection. A wrapper model is proposed in [7] to use a clustering algorithm in evaluating the quality of feature selection.

Recently, sparsity regularization, such as the $\ell_{2,1}$-norm of a matrix [5], in dimensionality reduction has been widely investigated and applied to feature selection including multi-task feature selection [1, 21], robust joint $\ell_{2,1}$-Norms [25], spectral feature selection [35], discriminative unsupervised feature selection [34]. Through sparsity regularization, feature selection can be embedded in the learning process.

The first attempt to select features on social media data is LinkedFS [29], a supervised algorithm. Various relations (coPost, coFollowing, coFollowed and Following) are extracted following social correlation theories [23]. LinkedFS significantly improves the performance of feature selection by incorporating these relations into feature selection. However, LinkedFS and LUFS are distinctively different. First, LinkedFS is formally stated as: $f : \{\mathbf{f}; \mathbf{X}, \mathbf{Y}\} \rightarrow \{\mathbf{f}'\}$ where $\mathbf{Y}$ contains the label information, while LUFS is a unsupervised feature selection algorithm. Second, LinkedFS exploits relations individually while LUFS employs relations as groups via social dimensions.

6. CONCLUSION

Linked data in social media presents new challenges to traditional feature selection algorithms, which assume the data instances to be independent and identically distributed. In this paper, we propose a novel unsupervised feature selection framework, LUFS, for linked data in social media. We utilize a recent developed concept of social dimensions from social network analysis to extract relations among linked data as groups, and define social dimension regularization inspired by Linear Discriminant Analysis to mathematically model these relations. We then propose the concept of pseudo-class labels to develop a new unsupervised feature selection framework by ensuring that instances in a social dimension are similar, and otherwise dissimilar. Experimental results on two datasets from real-world social media websites show that the proposed method can effectively exploit link information in comparison with the state-of-the-art unsupervised feature selection methods.

In social media networks, the availability of various link formation can lead to networks with relationships of different strengths [33, 28], which means that weak links and strong links are often mixed together. We plan to incorporate tie strength prediction into LUFS to further exploit link information. Also we believe that the concept of pseudo-class label introduced in the paper is a powerful means to effectively constrain the learning space of unsupervised feature selection and can be extended to different applications without labeled data but additional information.

Acknowledgments

The work is, in part, supported by ARO (#025071) and NSF (#0812551).

7. REFERENCES

[1] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. *NIPS*, 19:41, 2007.

[2] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge Univ Pr, 2004.

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

[4] C. Constantinopoulos, M. Titsias, and A. Likas. Bayesian feature and model selection for gaussian mixture models. *TPAMI*, pages 1013–1018, 2006.

[5] C. Ding, D. Zhou, X. He, and H. Zha. R 1-pca: rotational invariant l 1-norm principal component analysis for robust subspace factorization. In *Proceedings of the 23rd international conference on Machine learning*, pages 281–288. ACM, 2006.

[6] R. Duda, P. Hart, D. Stork, et al. *Pattern classification*, volume 2. wiley New York, 2001.

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

[8] J. G. Dy and C. E. Brodley. Feature subset selection and order identification for unsupervised learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 247–254, 2000.

[9] J. G. Dy and C. E. Brodley. Visualization and interactive feature selection for unsupervised data. In *KDD*, pages 360–364, 2000.

[10] J. G. Dy, C. E. Brodley, A. C. Kak, L. S. Broderick, and A. M. Aisen. Unsupervised feature selection applied to content-based retrieval of lung images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):373–378, 2003.

[11] E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5220, 2004.

[12] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002.

[13] M. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of Seventeenth International Conference on Machine Learning (ICML-00)*. Morgan Kaufmann Publishers, 2000.

[14] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. *NIPS*, 18:507, 2006.

[15] R. Horn and C. Johnson. *Matrix analysis*. Cambridge Univ Pr, 1990.

[16] G. John, R. Kohavi, and K. Pfleger. Irrelevant feature and the subset selection problem. In W. Cohen and H. H., editors, *Machine Learning: Proceedings of the Eleventh International Conference*, pages 121–129, New Brunswick, N.J., 1994. Rutgers University.

[17] Y. Kim, W. Street, and F. Menczer. Feature selection for unsupervised learning via evolutionary search. In *KDD*, pages 365–369, 2000.

[18] H. Liu and H. Motoda. *Computational methods of feature selection*. Chapman & Hall, 2008.

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

[20] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. on Knowledge and Data Engineering*, 17(3):1–12, 2005.

[21] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient l 2, 1-norm minimization. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 339–348. AUAI Press, 2009.

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

[23] P. Marsden and N. Friedkin. Network studies of social influence. *Sociological Methods and Research*, 22(1):127–151, 1993.

[24] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):26113, 2004.

[25] F. Nie, H. Huang, X. Cai, and C. Ding. Efficient and robust feature selection via joint l21-norms minimization. NIPS, 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*, pages 1226–1238, 2005.

[27] V. Roth and T. Lange. Feature selection in clustering problems. *NIPS*, 16:473–480, 2004.

[28] J. Tang, H. Gao, and H. Liu. mtrust: Discerning multi-faceted trust in a connected world. In *The ACM international conference on Web search and data mining*, 2012.

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

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

[31] X. Wang, L. Tang, H. Gao, and H. Liu. Discovering overlapping groups in social media. In *2010 IEEE International Conference on Data Mining*, pages 569–578. IEEE, 2010.

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

[33] R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 981–990. ACM, 2010.

[34] Y. Yang, H. Shen, Z. Ma, Z. Huang, and X. Zhou. L21-norm regularized discriminative feature selection for unsupervised learning. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[35] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1151–1157. ACM, 2007.

[36] Z. Zhao, L. Wang, and H. Liu. Efficient spectral feature selection with minimum redundancy. In *Proceedings of the Twenty-4th AAAI Conference on Artificial Intelligence (AAAI)*, 2010.