An Unsupervised Feature Selection Framework for Social Media Data

Jiliang Tang, Huan Liu, *Fellow, IEEE*

Abstract—The explosive usage of social media produces massive amount of unlabeled and high-dimensional data. Feature selection has been proven to be effective in dealing with high-dimensional data for efficient learning and data mining. Unsupervised feature selection remains a challenging task due to the absence of label information based on which feature relevance is often assessed. The unique characteristics of social media data further complicate the already challenging problem of unsupervised feature selection, e.g., social media data is inherently linked, which makes invalid the independent and identically distributed assumption, bringing about new challenges to unsupervised feature selection algorithms. In this paper, we investigate a novel problem of feature selection for social media data in an unsupervised scenario. In particular, we analyze the differences between social media data and traditional attribute-value data, investigate how the relations extracted from linked data can be exploited to help select relevant features, and propose a novel unsupervised feature selection framework, LUFS, for linked social media data. We systematically design and conduct systemic experiments to evaluate the proposed framework on datasets from real-world social media websites. The empirical study demonstrates the effectiveness and potential of our proposed framework.

Index Terms—Unsupervised Feature Selection, Linked data, Social Media, Pseudo Labels, Social Dimension Regularization

1 Introduction

In recent years, the rapid emergence of social media services such as Facebook and Twitter allows more and more users to participate in online social activities such as posting blogs or microblogs, uploading photos and connecting with other like-minded users. The explosive popularity of social media produces massive data at an unprecedented speed. For example, 250 million tweets are posted per day[1]; 3,000 photos are uploaded per minute to Flickr; and the number of Facebook users has increased from 100 million in 2008 to 800 million in 2011[2]. The massive and high-dimensional social media data challenges traditional data mining tasks such as classification and clustering due to curse of dimensionality and scalability issues. One traditional and effective approach to handle high-dimensional data is *feature selection* [8], [20], which aims to select a subset of relevant features from high-dimensional feature space that minimize redundancy and maximize relevance to the targets (e.g., class label). Feature selection helps improve the performance of learning models by alleviating the curse of dimensionality, speeding up the learning process, and improving the generalization capability of a learning model [18], [21].

Feature selection algorithms are broadly divided into *supervised* and *unsupervised* feature selection according to whether the training data is labeled or unlabeled. In supervised feature selection, the relevance of features is accessed by their capability of distinguishing different classes. It is time-consuming and costly to obtain labeled data and given the scale of unlabeled data in social media, we propose to study unsupervised feature selection. Without label information, unsupervised feature selection is particularly difficult due to the definition of relevancy of features becomes unclear [11], [10], [4], [1]. Furthermore, with high-dimensional data, it is likely to find many sets of features that seem equally good without considering additional constraints [11], [10]. Most existing feature selection algorithms work only with attribute-value data while social media data is inherently linked, adding further challenges to feature selection.

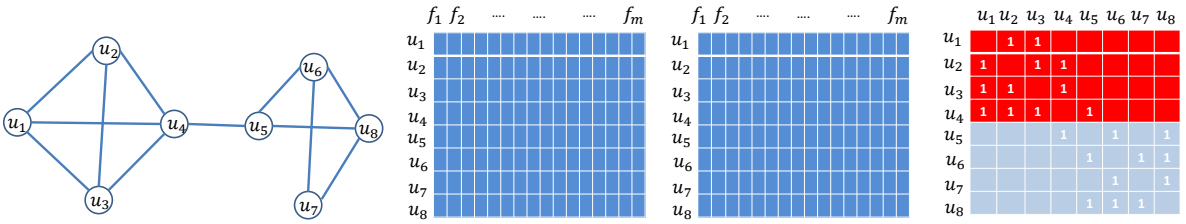
As we know, linked data provides link information beyond attribute value data. Figure 1 shows a simple example of linked data in social media and its two data representations. Figure 1(a) shows 8 linked instances (u_1 to u_8). These instances usually form groups and instances within groups have more connections than instances between groups [36]. 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 (or instance-feature matrix): 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) (or instance-instance matrix), in general, indicating the correlations among instances [35]. The availability of link information presents unprecedented opportunities to advanced research for feature selection.

Linked data in social media presents both challenges as well as new opportunities for unsupervised feature selection. In this paper, we study a novel problem of unsupervised feature selection for linked data. In particular, we investigate: (1) *how to exploit and model the relations among data instances*, and (2) *how*

- *J. Tang and H. Liu are with the Department of Computer Science, Arizona State University, Tempe, AZ, 85281. E-mail: {Jiliang.Tang, Huan.Liu}@asu.edu*
- *An shorter version of this submission was published in the proceeding of KDD2012 with the title "Unsupervised Feature Selection for linked Social Media Data" [33].*

1. http://techcrunch.com/2011/06/30/twitter-3200-million-tweets/

2. http://en.wikipedia.org/wiki/Facebook



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

Fig. 1. A Simple Example of Linked Social Media Data

to take advantage of these relations for feature selection in an unsupervised scenario. One key difference between supervised and unsupervised feature 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 [39]. In supervised learning, label information plays the role of constraint. Without labels, alternative constraints can be used, such as data variance and separability, for unsupervised learning [30], [8], [5], [16], [39], [4]. However, most of them evaluate the importance of a feature individually and neglect the feature correlation [40], [38]. Additionally, without considering additional constraints, these methods are likely to find many sets of features that seem equally good [20]. The availability of link information can provide more constraints for unsupervised feature selection and can potentially improve its performance. Furthermore, social theories are developed by sociologists to explain the formation of links in social media. For example, social correlation theories suggest that linked instances are likely to be similar. These social theories can help us exploit and model link information.

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:

- Introducing the concept of pseudo-class labels, enabling us to extract constraints from linked data, i.e., link information and attribute-value information, for unsupervised feature selection.
- Developing two approaches to exploit the individual and group behaviors of linked instances via graph regularization and social dimension regularization, separately.
- 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 approaches to capture link information, 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 $(\mathbf{A}, \mathbf{B}, \ldots)$. $\mathbf{A}(i,:)$ and $\mathbf{A}(:,j)$ denote the i^{th} row and j^{th} column of $\mathbf{A}$, respectively.

Let $\mathbf{p} = \{p_1, p_2, \ldots, p_n\}$ be the set of instances in social media (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 (matrix) representation of $\mathbf{p}$ w.r.t. $\mathbf{f}$ where $\mathbf{x}_i(j)$ is the feature value of f_j in p_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 in the form of link information. Let $\mathbf{R} \in \mathbb{R}^{n \times n}$ be their link representation where $\mathbf{R}(i,j) = 1$ if p_i and p_j are linked, otherwise zero (e.g., the right subgraph in Figure 1(c)). In this work, we assume that the relations among data instance is undirected, i.e., $\mathbf{R} = \mathbf{R}^\top$, however, the extension to directed relations is straightforward.

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 $\mathcal{F}$, which can select a subset of relevant features $\mathbf{f}'$ from $\mathbf{f}$ with size k by utilizing both $\mathbf{X}$ and $\mathbf{R}$ for these n linked instances in an unsupervised scenario, formally stated as,

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

while the traditional task of unsupervised feature selection, formally defined as,

$$\mathcal{F} : \{\mathbf{f}; \mathbf{X}\} \rightarrow \{\mathbf{f}'\}. \quad (2)$$

Our proposed task substantially differs from traditional unsupervised feature selection problem - linked data provides link information beyond attribute-value data and in general, link information reveals the correlation between instances, enabling advanced research for feature selection.

3 Unsupervised Feature Selection Framework: LUFS

Let $\mathbf{s} = \pi(\overbrace{0,\ldots,0}^{m-k},\overbrace{1,\ldots,1}^{k})$, 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 $\text{diag}(\mathbf{s})\mathbf{X}$ with k selected features, where $\text{diag}(\mathbf{s})$ 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 feature selection. 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 $\mathbf{Y} = \mathbf{W}^\top \text{diag}(\mathbf{s})\mathbf{X} \in \mathbb{R}^{c\times n}$. $\mathbf{Y}_{ji}$ indicates the likelihood of the i-th instance belonging to the j-th class. Following the widely adopted constraints on the cluster indicator matrix, we add orthogonal constraints on $\mathbf{Y}$. Since $\mathbf{X}$ is centered, it is easy to verify that $\mathbf{Y}$ is also centered,

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

Both supervised and unsupervised feature selection tasks aim to solve the same problem: selecting features consistent with given constraints. In the supervised setting, label information plays the role of constraints; in the unsupervised setting, pseudo-class label information is tantamount to the provided label information and we seek pseudo-class label information from linked social media data. In particular, we seek constraints from both link information $\mathbf{R}$ and attribute-value part $\mathbf{X}$ to fit pseudo-class label. For each type of information, we provide both an intuitive way and an advanced way to extract constraints for feature selection in this paper. The reasons are two-fold. First, feature selection for linked data is a relatively novel problem; we try to explore it more widely to seek a better and deeper understanding of this problem. Second, we try to investigate both intuitive and recently proposed techniques in the studied problem to help us understand their advantages and disadvantages in the proposed problem.

For link information, a widely adopted assumption is that linked instances are likely to share similar labels, which can be explained by social correlation theories such as homophily [25] and social influence [24]. Based on this intuition, we propose graph regularization to capture link information based on social correlation theories. Users in social networks are likely to form groups and users within groups are similar while users from different groups are dissimilar, which is supported by social dimension assumption [34]. An advanced way social dimension regularization based on social dimension assumption is proposed to extract constraints from link information as well.

For attribute-value information, the intuitive way is spectral analysis based on the assumption that instances with similar content should have similar labels. Recently, local discriminate analysis is introduced in feature selection for attribute-value data and promising results are reported [38]. These motivate us to capture attribute-value information through spectral analysis or discriminate analysis. Considering different combinations of components for capturing link information and attribute-value information, there are four variants of the proposed framework LUFS, i.e., LUFS-SG, LUFS-SSD, LUFS-DG and LUFS-DSD, as shown in Table 1. More details will be introduced in the following subsections.

3.1 Capturing Link Information

In general, linked instances are correlated and allow us to exploit their correlations for feature selection. In following subsections, by considering their individual and group behaviors, we introduce graph regularization and social dimension regularization to capture the dependency among linked instances as individuals and groups, respectively.

3.1.1 Graph Regularization for Link Information

Two linked instances are distinct from two instances of attribute-value data due to the correlations between them [31]. For example, two linked users in Twitter are more likely to have similar interests than two randomly picked users. These correlations can be explained by social correlation theories such as homophily [25] and social influence [24]. Homophily suggests that two instances with similar topics are more likely to be linked while social influence theory indicates that two linked instances are more likely to have similar topics.

Social correlation theories suggest that two linked instances in social media are more likely to have similar labels, which can be exploited through pseudo-class label. For p_i and p_j, if linked, we want to their labels, i.e., $\mathbf{Y}(:,i)$ and $\mathbf{Y}(:,j)$, are similar. Therefore, we *force the labels of two linked data instances close to each other* via minimizing the following term of graph regularization,

$$\begin{aligned}
&\frac{1}{2}\sum_i\sum_j \mathbf{R}(i,j)\|\mathbf{Y}(:,i)-\mathbf{Y}(:,j))\|_2^2,\\
&=\frac{1}{2}\sum_k\sum_i\sum_j \mathbf{R}(i,j)(\mathbf{Y}(k,i)-\mathbf{Y}(k,j))^2,\\
&=\sum_k \mathbf{Y}(k,:)(\mathbf{D}_R-\mathbf{R})\mathbf{Y}(k,:),\\
&=Tr(\mathbf{Y}\mathbf{L}_R\mathbf{Y}^\top),
\end{aligned} \tag{3}$$

TABLE 1
Four Variants of LUFS

		Link Information	
		Graph Regularization (G)	Social Dimension Regularization (SD)
Attribute-Value Information	Spectral Analysis (S)	LUFS-SG	LUFS-SSD
	Discriminative Analysis (D)	LUFS-DG	LUFS-DSD

where $\mathbf{L}_R = \mathbf{D}_R - \mathbf{R}$ is the Laplacian matrix based on the link information $\mathbf{R}$, and $\mathbf{D}_R$ is a diagonal matrix with $\mathbf{D}_R(i,i) = \sum_j \mathbf{R}(j,i)$.

3.1.2 Social Dimensions for Link Information

In [34], social dimension is introduced as a means to integrate the interdependency among linked data and attribute-value data. Social dimension can capture group behaviors of linked instances: 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 [34].

Social dimension extraction is a well-studied problem by social network analysis community [26], [12], [34]. In our work, we adopt a widely used algorithm, i.e., Modularity Maximization [26], 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 $\mathbf{H} \in \mathbb{R}^{n\times K}$ 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 p_i.

It is easy to verify that the optimal $\mathbf{H}$ is the top-K eigenvectors of the modularity matrix $\mathbf{M}$. $\mathbf{H}$ is continuous and is mixed with positive and negative values. Similar to spectral clustering to obtain the cluster labels, the standard K-means algorithm is performed on $\mathbf{H}$ to obtain the discrete-valued social dimension assignment , $\hat{\mathbf{H}}(i,j) = 1$, if p_i is in the j-th dimension, and $\hat{\mathbf{H}}(i,j) = 0$, 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 while have fewer connections between social dimensions. The discrete-valued social dimension indicator matrix, $\hat{\mathbf{H}}$, is:

$$\hat{\mathbf{H}}^\top = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix},$$

Since social dimensions represent a type of user affiliations, inspired by Linear Discriminate Analysis, we define three matrices: within, between, and total social dimension scatter matrices $\mathbf{S}_w$, $\mathbf{S}_b$, 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 $\mathbf{S}_w + \mathbf{W}_b = \mathbf{S}_t$. $\mathbf{F}$ is the weighted social dimension indicator matrix, which can be obtained from $\hat{\mathbf{H}}$ as below,

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

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 maximizing the following term,

$$Tr\big((\mathbf{S}_t)^{-1}\mathbf{S}_b\big) = Tr\big((\mathbf{Y}\mathbf{Y}^\top)^{-1}\mathbf{Y}\mathbf{F}\mathbf{F}^\top\mathbf{Y}^\top\big). \tag{9}$$

3.2 Capturing Attribute-Value Information

With the pseudo-class label, we are further allowed to capture information from attribute-value part in a supervised manner. In the following subsections, we will introduce two methods to exploit attribute-value information for feature selection based on two recently developed techniques, i.e., spectral analysis [23] and discriminative analysis [38].

3.2.1 Spectral Analysis for Attribute-Value Information

Spectral analysis indicates that similar data instances should have similar labels. Assume that $\mathbf{S} \in \mathbb{R}^{n\times n}$ is the similarity matrix based on $\mathbf{X}$, obtained through a RBF kernel in this work as,

$$\mathbf{S}(i,j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}}. \tag{10}$$

We force the pseudo-class labels of similar data instances to be close to each other. Therefore the constraint from attribute-value part can be formulated

to minimize the following term,

$$\begin{aligned}&\frac{1}{2}\sum_i\sum_j \mathbf{S}(i,j)\|\mathbf{Y}(:,i)-\mathbf{Y}(:,j))\|_2^2,\\&=\frac{1}{2}\sum_k\sum_i\sum_j \mathbf{S}(i,j)(\mathbf{Y}(k,i)-\mathbf{Y}(k,j))^2,\\&=\sum_k \mathbf{Y}(k,:)(\mathbf{D}_S-\mathbf{S})\mathbf{Y}(k,:),\\&=Tr(\mathbf{Y}\mathbf{L}_S\mathbf{Y}^\top),\end{aligned} \tag{11}$$

where $\mathbf{L}_S = \mathbf{D}_S - \mathbf{S}$ is a Laplacian matrix and $\mathbf{D}_S$ is a diagonal matrix with its elements defined as $\mathbf{D}_S(i,i) = \sum_{j=1}^{n} \mathbf{S}(i,j)$.

3.2.2 Discriminative Analysis for Attribute-Value Information

Another advantage from the introducing of the pseudo-class label is that it allows us to exploit local discriminative information [38] for attribute-value part. For each data point p_j, we obtain its K' nearest neighbors, denoted as $\mathcal{U}^j = \{j_1, j_2, \ldots, j_{K'}\}$. Let $\mathbf{N}^j = [\mathbf{x}_j, \mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_{K'}}]$ and $\mathbf{Y}^j = [\mathbf{Y}(:,j), \mathbf{Y}(:,j_1), \ldots, \mathbf{Y}(:,j_{K'})]$ be the local data and local label matrix, respectively. It is easy to verify that $\mathbf{Y}^j = \mathbf{Y}\mathbf{Z}^j$ where $\mathbf{Z}^j \in \{0,1\}^{n\times(K'+1)}$ is defined as,

$$\mathbf{Z}^j(p,q) = \begin{cases} 1 & \text{if } p = \{\hat{\mathcal{U}}^j(q)\},\\ 0 & \text{otherwise.}\end{cases} \tag{12}$$

where $\hat{\mathcal{U}}^j = \{j, \mathcal{U}^j\}$. Inspired by Linear Discriminate Analysis, the local total scatter matrix $\mathbf{T}^j$ and between class scatter matrix $\mathbf{B}^j$ for $\mathbf{x}_j$ are defined as below,

$$\begin{aligned}\mathbf{T}^j &= \mathbf{N}^j\mathbf{P}(\mathbf{N}^j)^\top,\\ \mathbf{B}^j &= \mathbf{N}^j\mathbf{P}(\mathbf{Y}^j)^\top\mathbf{Y}^j\mathbf{P}(\mathbf{N}^j)^\top,\end{aligned} \tag{13}$$

where $\mathbf{P} = \mathbf{I} - \frac{1}{K'+1}\mathbf{1}_{K'+1}\mathbf{1}_{K'+1}^\top \in \mathbb{R}^{(K'+1)\times(K'+1)}$. It is easy to verify that $\mathbf{P} = \mathbf{P}^\top = \mathbf{P}^\top\mathbf{P}$. Local discriminative information for p_j captures the discriminative information among p_j and its neighbors. With the definition of $\mathbf{T}^j$ and $\mathbf{B}^j$ and analogous to Linear Discriminate Analysis, local discriminative score, DS_j for $\mathbf{x}_j$ is defined as,

$$DS_j = Tr\big((\mathbf{T}^j + \eta\mathbf{I})^{-1}\mathbf{B}^j\big). \tag{14}$$

where $\eta\mathbf{I}$ is added to make the term $\mathbf{T}^j + \eta\mathbf{I}$ invertible. Apparently, a larger DS_j suggests a higher discriminative ability w.r.t. the datum of $\mathbf{x}_j$. Therefore local discriminative information of attribute-value part can be captured via obtaining the highest discriminative scores for all data instances, which can be formulated to maximize: $\sum_{j=1}^{n} DS_j$. After some derivations, the maximization problem can be converted into the following minimization problem [38],

$$\min_{\mathbf{W}} \quad Tr(\mathbf{Y}\mathbf{G}\mathbf{Y}^\top), \qquad s.t. \quad \mathbf{Y}\mathbf{Y}^\top = \mathbf{I} \tag{15}$$

where $\mathbf{G}$ is defined as,

$$\mathbf{G} = \sum_{j=1}^{n}\big(\mathbf{Z}^j\mathbf{P}(\mathbf{N}^j\mathbf{P}(\mathbf{N}^j)^\top + \eta\mathbf{I})^{-1}\mathbf{P}(\mathbf{Z}^j)^\top\big). \tag{16}$$

3.3 Unsupervised Feature Selection Framework

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 to fit the introduced pseudo-class labels. Considering different combinations of components for capturing link information and attribute-value information, we propose four variants of LUFS, i.e., LUFS-SG, LUFS-SSD, LUFS-DG and LUFS-DSD, as shown in Table 1. The optimization problems for these four variants are stated in Table 2.

Since $\mathbf{Y}\mathbf{Y}^\top = \mathbf{I}$, $\mathbf{S}_t$ 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 a minimization problem,

$$\min Tr(\mathbf{Y}\mathbf{Y}^\top) - Tr(\mathbf{Y}\mathbf{F}\mathbf{F}^\top\mathbf{Y}^\top). \tag{17}$$

The reason for adding the constant in social dimension regularization will be explained later.

The constraint on s makes problem in Table 2 mixed integer programming [3], which is still difficult to solve. We observe that diag(s) and $\mathbf{W}$ is as the form of diag(s)$\mathbf{W}$ in $\mathbf{Y}$. Since s is a binary vector and $m-k$ rows of the diag(s) are all zeros, diag(s)$\mathbf{W}$ is a matrix where the elements of many rows are all zeros. This motivates us to absorb the diag(s) into $\mathbf{W}$, $\mathbf{W} = \text{diag(s)}\mathbf{W}$, and add $\ell_{2,1}$ norm on $\mathbf{W}$ to ensure the sparsity of $\mathbf{W}$ in rows and achieve feature selection. The problems for LUFS-SG, LUFS-SSD, LUFS-DG and LUFS-DSD after relaxation are shown in Table 3.

Since $\mathbf{X}$ is high dimensional, $\mathbf{X}\mathbf{X}^\top$ might be singular. We add $\lambda\mathbf{I}$ to ($\mathbf{X}\mathbf{X}^\top$ to make $\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I}$) nonsingular and this setting is very common in practice such as the standard linear regression problem and the linear discriminate analysis. $\|\mathbf{W}\|_{2,1}$ 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 [6],

$$\|\mathbf{W}\|_{2,1} = \sum_{i=1}^{m}\sqrt{\sum_{j=1}^{k}\mathbf{W}^2(i,j)} = \sum_{i=1}^{m}\|\mathbf{W}(i,:)\|_2. \tag{18}$$

Next we will develop a theorem to show that LUFS-SG, LUFS-SSD, LUFS-DG and LUFS-DSD can be converted into a unique form after some direct derivations, beginning with the following lemma.

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 $\mathbf{I}_n - \mathbf{E}\mathbf{E}^\top$ is a symmetric and positive semi-definite matrix.*

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

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

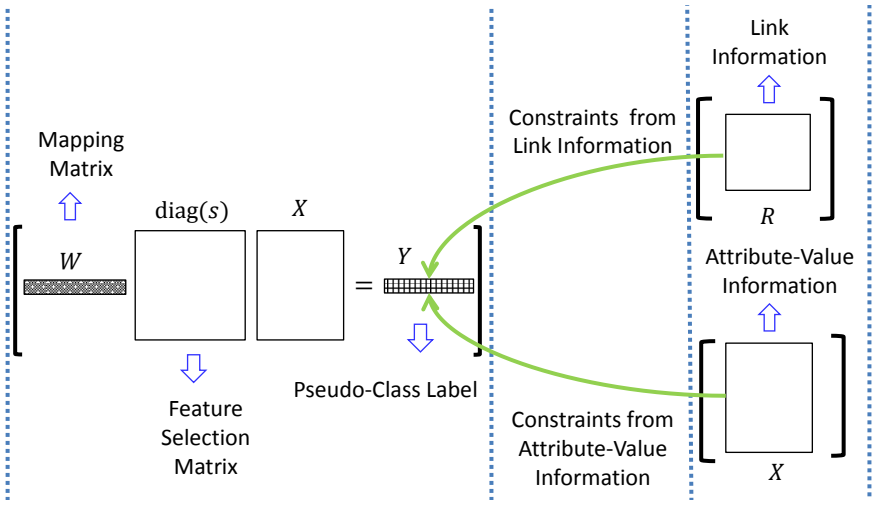


Fig. 2. Illustration of the LUFS Framework

TABLE 2
Optimization Problems for the Four Variants of LUFS

LUFS-SG	$\min_{\mathbf{W},\mathbf{s}} \; Tr(\mathbf{Y}\mathbf{L}_S\mathbf{Y}^\top) + \alpha Tr(\mathbf{Y}\mathbf{L}_R\mathbf{Y}^\top) \quad s.t.\ \mathbf{s}\in\{0,1\}^n,\ \mathbf{s}^\top\mathbf{1}_n = k,\ \ \mathbf{Y}\mathbf{Y}^\top = \mathbf{I}.$
LUFS-SSD	$\min_{\mathbf{W},\mathbf{s}} \; Tr(\mathbf{Y}\mathbf{L}_S\mathbf{Y}^\top) - \alpha Tr\big((\mathbf{S}_t)^{-1}\mathbf{S}_b\big) \quad s.t.\ \mathbf{s}\in\{0,1\}^n,\ \mathbf{s}^\top\mathbf{1}_n = k,\ \ \mathbf{Y}\mathbf{Y}^\top = \mathbf{I}.$
LUFS-DG	$\min_{\mathbf{W},\mathbf{s}} \; Tr(\mathbf{Y}\mathbf{G}\mathbf{Y}^\top) + \alpha Tr(\mathbf{Y}\mathbf{L}_R\mathbf{Y}^\top) \quad s.t.\ \mathbf{s}\in\{0,1\}^n,\ \mathbf{s}^\top\mathbf{1}_n = k,\ \ \mathbf{Y}\mathbf{Y}^\top = \mathbf{I}.$
LUFS-DSD	$\min_{\mathbf{W},\mathbf{s}} \; Tr(\mathbf{Y}\mathbf{G}\mathbf{Y}^\top) - \alpha Tr\big((\mathbf{S}_t)^{-1}\mathbf{S}_b\big) \quad s.t.\ \mathbf{s}\in\{0,1\}^n,\ \mathbf{s}^\top\mathbf{1}_n = k,\ \ \mathbf{Y}\mathbf{Y}^\top = \mathbf{I}.$

TABLE 3
Optimization Problems for the Four Variants of LUFS after Relaxation

LUFS-SG	$\min_{\mathbf{W}} \; Tr(\mathbf{W}^\top\mathbf{X}\mathbf{L}_S\mathbf{X}^\top\mathbf{W}) + \beta\|\mathbf{W}\|_{2,1} + \alpha Tr(\mathbf{W}^\top\mathbf{X}\mathbf{L}_R\mathbf{X}^\top\mathbf{W})$ s.t. $\mathbf{W}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})\mathbf{W} = \mathbf{I}_c,$
LUFS-SSD	$\min_{\mathbf{W}} \; Tr(\mathbf{W}^\top\mathbf{X}\mathbf{L}_S\mathbf{X}^\top\mathbf{W} + \beta\|\mathbf{W}\|_{2,1} + \alpha Tr(\mathbf{W}^\top\mathbf{X}(\mathbf{I}_n - \mathbf{F}\mathbf{F}^\top)\mathbf{X}^\top\mathbf{W})$ s.t. $\mathbf{W}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})\mathbf{W} = \mathbf{I}_c,$
LUFS-DG	$\min_{\mathbf{W}} \; Tr(\mathbf{W}^\top\mathbf{X}\mathbf{G}\mathbf{X}^\top\mathbf{W} + \beta\|\mathbf{W}\|_{2,1} + \alpha Tr(\mathbf{W}^\top\mathbf{X}\mathbf{L}_R\mathbf{X}^\top\mathbf{W}))$ s.t. $\mathbf{W}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})\mathbf{W} = \mathbf{I}_c,$
LUFS-DSD	$\min_{\mathbf{W}} \; Tr(\mathbf{W}^\top\mathbf{X}\mathbf{G}\mathbf{X}^\top\mathbf{W}) + \beta\|\mathbf{W}\|_{2,1} + \alpha Tr(\mathbf{W}^\top\mathbf{X}(\mathbf{I}_n - \mathbf{F}\mathbf{F}^\top)\mathbf{X}^\top\mathbf{W})$ s.t. $\mathbf{W}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})\mathbf{W} = \mathbf{I}_c,$

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,

$$\mathbf{E}^\top\mathbf{E} = \mathbf{I}_m \rightarrow \mathbf{V}\Sigma^\top\Sigma\mathbf{V}^\top = \mathbf{V}\mathbf{V}^\top$$
$$\rightarrow 1 \geq \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_m \geq 0. \tag{19}$$

Then,

$$\mathbf{I}_n - \mathbf{E}\mathbf{E}^\top = \mathbf{U}(\mathbf{I}_n - \Sigma\Sigma^\top)\mathbf{U}^\top. \tag{20}$$

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 semi-definite matrix. Thus $\mathbf{U}(\mathbf{I}_n - \Sigma\Sigma^\top)\mathbf{U}^\top$ is a semi-definite matrix, which completes the proof. □

The importance of Lemma 1 is two-fold. First, it explains why we need a constant on social dimension regularization. Adding the constant can guarantee the convexity of the objective functions for LUFS-SSD and LUFS-DSD. Second, it paves the way for the following theorem for LUFS.

Theorem 3.1: Problems in Table 3 can be converted into the following unique optimization problem,

$$\min_{\mathbf{W}} \quad f(\mathbf{W}) = Tr(\mathbf{W}^\top\mathbf{A}\mathbf{W}) + \beta\|\mathbf{W}\|_{2,1},$$
$$s.t. \quad \mathbf{W}^\top\mathbf{B}\mathbf{W} = \mathbf{I}_c, \tag{21}$$

where $\mathbf{A}$ is a symmetric and positive semi-definite matrix and $\mathbf{B}$ is a symmetric and positive matrix.

Proof: We will use LUFS-SSD as an example to demonstrate the proof process. According to above analysis, LUFS-SSD is to solve the following minimization problem,

$$\min_{\mathbf{W}} \quad Tr(\mathbf{W}^\top\mathbf{X}\mathbf{L}_S\mathbf{X}^\top\mathbf{W} + \beta\|\mathbf{W}\|_{2,1}$$
$$+ \alpha Tr(\mathbf{W}^\top\mathbf{X}(\mathbf{I}_n - \mathbf{F}\mathbf{F}^\top)\mathbf{X}^\top\mathbf{W})$$
$$s.t. \quad \mathbf{W}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})\mathbf{W} = \mathbf{I}_c. \tag{22}$$

It suffices to show how to construct a symmetric and positive semi-definite matrix $\mathbf{A}$ and a symmetric and positive matrix $\mathbf{B}$ from Eq. (22) for LUFS-SSD.

We construct $\mathbf{B} = \mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I}$. It is easy to check that $\mathbf{B}$ is symmetric and positive when $\lambda \neq 0$ and then the constraint in Eq. (22) is converted into the one in Eq. (21).

We construct $\mathbf{A} = \mathbf{X}\mathbf{L}_S\mathbf{X}^\top + \alpha\mathbf{X}(\mathbf{I}_n - \mathbf{F}\mathbf{F}^\top)\mathbf{X}^\top$ and then the objective function in Eq. (22) is converted into the one in Eq. (21). According to Lemma 1, $(\mathbf{I}_n - \mathbf{F}\mathbf{F}^\top)$ is symmetric and positive semi-definite; and according to the definition of Laplacian matrix, $\mathbf{X}\mathbf{L}\mathbf{X}^\top$ is symmetric and positive semi-definite. Thus, $\mathbf{A}$ is symmetric and positive semi-definite.

Similar Process can be applied to LUFS-SG, LUFS-DG and LUFS-DSD. To save space, we ignore the

details and summarize the definitions of $\mathbf{A}$ and $\mathbf{B}$ for LUFS-SG, LUFS-DG and LUFS-DSD as follows:

$$\text{LUFS-SG}: \mathbf{A} = \mathbf{X}\mathbf{L}_S\mathbf{X}^\top + \alpha\mathbf{X}\mathbf{L}_R\mathbf{X}^\top, \mathbf{B} = \mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I},$$
$$\text{LUFS-DG}: \mathbf{A} = \mathbf{X}\mathbf{G}\mathbf{X}^\top + \alpha\mathbf{X}(\mathbf{I}_n - \mathbf{F}\mathbf{F}^\top)\mathbf{X}^\top,$$
$$\mathbf{B} = \mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I},$$
$$\text{LUFS-DSD}: \mathbf{A} = \mathbf{X}\mathbf{G}\mathbf{X}^\top + \alpha\mathbf{X}\mathbf{L}_R\mathbf{X}^\top, \mathbf{B} = \mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I}.$$

which completes the proof. □

3.4 Optimization Algorithm for LUFS

In recent years, many methods have been proposed to solve the $\ell_{2,1}$-norm minimization problem [22], [27], [38]. However, our problem is different from these existing ones due to the orthogonal constraint in Eq. (21). Hence, we propose the following algorithm, as shown in Algorithm 2, to optimize the problem in Eq. (21).

Algorithm 1 LUFS

Input: $\{\mathbf{X}, \mathbf{R}, \alpha, \beta, \lambda, c, K, k\}$
Output: k most relevant features

1: Construct $\mathbf{A}$ according to the components chosen for LUFS
2: Set $\mathbf{B} = \mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I}$
3: Set $t = 0$ and initialize $\mathbf{D}_0$ as an identity matrix
4: **while** Not convergent **do**
5: Set $\mathbf{C}_t = \mathbf{B}^{-1}(\mathbf{A} + \beta\mathbf{D}_t)$
6: 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
7: Update the diagonal matrix $\mathbf{D}_{t+1}$, where the i-th diagonal element is $\frac{1}{2\|\mathbf{W}_t(i,:)\|_2}$;
8: Set $t = t + 1$
9: **end while**
10: Sort each feature according to $\|\mathbf{W}(i,:)\|_2$ in **descending** order and select the top-k ranked ones;

Next we briefly review Algorithm 2. In line 1, according to the components chosen for LUFS, we construct $\mathbf{A}$ referring to Theorem 3.1. For example, we do the following steps to construct $\mathbf{A}$ for LUFS-SSD,

- Obtain the social dimension indicator matrix $\mathbf{H}$,
- Set $\mathbf{F} = \mathbf{H}(\mathbf{H}^\top\mathbf{H})^{-\frac{1}{2}}$,
- Construct $\mathbf{S}$ through RBF kernel and set $\mathbf{L} = \mathbf{D}_S - \mathbf{S}$,
- Construct $\mathbf{A} = \mathbf{X}\mathbf{L}\mathbf{X}^\top + \alpha\mathbf{X}(\mathbf{I}_n - \mathbf{F}\mathbf{F}^\top)\mathbf{X}^\top$.

the iterative algorithm to optimize Eq. (21) is presented from line 4 to line 9. 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 semi-definite 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:*

$$\min_{\mathbf{W}} \quad Tr(\mathbf{W}^\top\mathbf{A}\mathbf{W}), \qquad s.t. \quad \mathbf{W}^\top\mathbf{B}\mathbf{W} = \mathbf{I}_c, \tag{23}$$

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

Proof: Since $\mathbf{B}$ is symmetric and positive, let $\mathbf{B} = \mathbf{U}\Sigma\mathbf{U}^\top$ be the Singular Value Decomposition (SVD) of $\mathbf{B}$. It follows that $\mathbf{B} = \mathbf{U}\Sigma^{\frac{1}{2}}\Sigma^{\frac{1}{2}}\mathbf{U}^\top = \mathbf{E}\mathbf{E}^\top$ where $\mathbf{E} = \mathbf{U}\Sigma^{\frac{1}{2}}$. The objective function in Eq. (23) can be rewritten as,

$$\begin{aligned} &Tr(\mathbf{W}^\top\mathbf{A}\mathbf{W}), \\ &= Tr(\mathbf{W}^\top\mathbf{E}\mathbf{E}^{-1}\mathbf{A}(\mathbf{E}^{-1})^\top\mathbf{E}^\top\mathbf{W}), \\ &= Tr\big((\mathbf{W}^\top\mathbf{E})\mathbf{E}^{-1}\mathbf{A}(\mathbf{E}^{-1})^\top(\mathbf{E}^\top\mathbf{W})\big). \end{aligned} \tag{24}$$

Set $\mathbf{E}^\top\mathbf{W} = \hat{\mathbf{W}}$ and $\hat{\mathbf{W}}^\top\hat{\mathbf{W}}^\top = \mathbf{W}^\top\mathbf{B}\mathbf{W} = \mathbf{I}_c$, then Eq. (23) can be rewritten as,

$$\min_{\hat{\mathbf{W}}} \quad Tr(\hat{\mathbf{W}}^\top\mathbf{E}^{-1}\mathbf{A}(\mathbf{E}^{-1})^\top\hat{\mathbf{W}}), \qquad s.t. \quad \hat{\mathbf{W}}\hat{\mathbf{W}} = \mathbf{I}_c, \tag{25}$$

According to *Ky Fan Theorem* [17], the optimal $\hat{\mathbf{W}}$ is given by the smallest c singular value of $\mathbf{E}^{-1}\mathbf{A}(\mathbf{E}^{-1})^\top$. We assume that $\mathbf{g}_i$ is the i-th eigenvector of $\mathbf{E}^{-1}\mathbf{A}(\mathbf{E}^{-1})^\top$ corresponding to the i-th smallest eigenvalue and we have,

$$\begin{aligned} &\mathbf{E}^\top\mathbf{W} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_c] \\ &\Rightarrow \mathbf{W} = (\mathbf{E}^{-1})^\top[\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_c] \end{aligned} \tag{26}$$

It is easy to verify that $(\mathbf{E}^{-1})^\top\mathbf{g}_i$ is the i-th eigenvector of $\mathbf{B}^{-1}\mathbf{A}$ corresponding to the i-th smallest eigenvalue. Thus the optimal $\mathbf{W}$ consists of the eigenvectors of $\mathbf{B}^{-1}\mathbf{A}$ corresponding to the c smallest eigenvalues, which completes the proof. □

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

$$\sum_i \|\mathbf{w}_{t+1}^i\|_2 - \sum_i \frac{\|\mathbf{w}_{t+1}^i\|_2}{2\|\mathbf{w}_t^i\|_2} \leq \sum_i \|\mathbf{w}_t^i\|_2 - \sum_i \frac{\|\mathbf{w}_t^i\|_2^2}{2\|\mathbf{w}_t^i\|_2}. \tag{27}$$

Proof: According to [27], [31], the following inequality holds for any positive constants a and b,

$$\sqrt{a} - \frac{a}{2\sqrt{b}} \leq \sqrt{b} - \frac{b}{2\sqrt{b}} \tag{28}$$

Use $\|\mathbf{w}_{t+1}^i\|_2$ and $\|\mathbf{w}_t^i\|_2^2$ to replace a and b in Eq. (28), respectively, we can see that the following inequality holds for any i.

$$\|\mathbf{w}_{t+1}^i\|_2 - \frac{\|\mathbf{w}_{t+1}^i\|_2}{2\|\mathbf{w}_t^i\|_2} \leq \|\mathbf{w}_t^i\|_2 - \frac{\|\mathbf{w}_t^i\|_2^2}{2\|\mathbf{w}_t^i\|_2} \tag{29}$$

By summing Eq. (29) over i, Eq. (27) holds, which completes the proof. □

With the above two lemmas, we develop the following theorem regarding the convergence of Algorithm 2.

Theorem 3.2: At each iteration of Algorithm 2, the value of the objective function in Eq. (21) monotonically decreases.

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

$$\mathbf{W}_{t+1} = \min_{\mathbf{W}^\top \mathbf{B}\mathbf{W}=\mathbf{I}} Tr\big(\mathbf{W}^\top(\mathbf{A}+\beta\mathbf{D}_t)\mathbf{W}\big), \quad (30)$$

which indicates that,

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

That is to say,

$$\begin{aligned} &Tr(\mathbf{W}_{t+1}^\top\mathbf{A}\mathbf{W}_{t+1}) + \beta\sum_i \frac{\|\mathbf{W}_{t+1}(i,:)\|_2^2}{2\|\mathbf{W}_t(i,:)\|_2} \\ &\leq Tr(\mathbf{W}_t^\top\mathbf{A}\mathbf{W}_t) + \beta\sum_i \frac{\|\mathbf{W}_t(i,:)\|_2^2}{2\|\mathbf{W}_t^(i,:)\|_2}. \end{aligned} \quad (31)$$

Then we have the following inequality,

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

Meanwhile, according to Lemma 3, we have,

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

Therefore, we have the following inequality:

$$\begin{aligned} &Tr(\mathbf{W}_{t+1}^\top\mathbf{A}\mathbf{W}_{t+1}) + \beta\|\mathbf{W}_{t+1}\|_{2,1} \leq \\ &\qquad Tr(\mathbf{W}_t^\top\mathbf{A}\mathbf{W}_t) + \beta\|\mathbf{W}_t\|_{2,1}, \end{aligned} \quad (32)$$

which completes the proof. □

According to Theorem 3.2, Algorithm 2 converges to the optimal $\mathbf{W}$ for the problem in Eq. (21). Next, we will analyze the time complexity of Algorithm 2.

Time Complexity : there are three main computations in Algorithm 2 - computing $\mathbf{A}$, computing $\mathbf{B}$ and computing the c eigenvectors of $\mathbf{C}_t$. For four variants of LUFS, the time complexities of computing $\mathbf{B}$ and computing the c eigenvectors of $\mathbf{C}_t$ is the same as,

- The major operation of constructing $\mathbf{B}$ is $\mathbf{X}\mathbf{X}^\top$, whose time complexity is about $O(n^2m)$.
- The most time-consuming operation of updating $\mathbf{W}$ is to obtain the c eigenvectors of $\mathbf{C}_t \in \mathbb{R}^{m\times m}$ corresponding to the first c smallest eigenvalues. The time complexity of this operation achieved by Lanczos algorithm is $O(cm^2)$.

However, the time complexities of computing $\mathbf{A}$ for different variants are different. The time complexity to compute $\mathbf{A}$ contains two parts. The first part is to compute $\mathbf{X}\mathbf{L}_S\mathbf{X}^\top$, $\mathbf{X}\mathbf{G}\mathbf{X}^\top$, $\mathbf{X}\mathbf{L}_R\mathbf{X}^\top$ or $\mathbf{X}(\mathbf{I}_n - \mathbf{F}\mathbf{F}^\top)\mathbf{X}^\top$. The time complexity of this part is the same for all variants as $O(n^2m)$. The second part is to calculate $\mathbf{L}_S$, $\mathbf{G}$, $\mathbf{L}_R$ or $\mathbf{I}_n - \mathbf{F}\mathbf{F}^\top$, which needs $O(n^2m)$, $O(K'n^2)$, $O(n)$ [3] and $O(Kn^2+cKn+cn^2)$, respectively. Here we can conclude that for link information, the time complexity of graph regularization $O(n)$ is lower than that of social dimension regularization $O(Kn^2+cKn+cn^2)$, while for attribute-value information, the time complexity of discriminative analysis $O(K'n^2)$ is lower than that of spectral analysis $O(n^2m)$. Therefore the operations that we need to compute the second part of four variants satisfy LUFS-SSD > LUFS-SG > LUFS-DSD > LUFS-DG.

TABLE 4
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
Clustering Coefficient	0.1224	0.3301

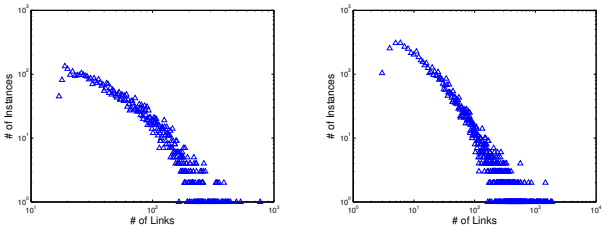
4 Experiments and Discussion

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 compare the four variants of LUFS, and evaluate the effectiveness of LUFS 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

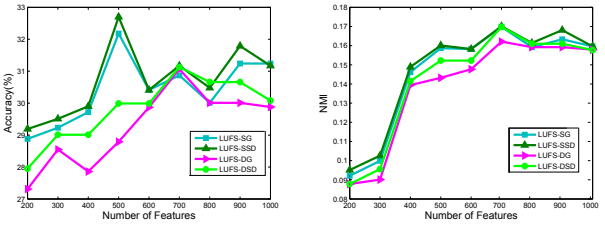
We collect two datasets from real-world social media websites, i.e., BlogCatalog[4] and Flickr[5], which are the subsets of two public available datasets used in [36] to uncover overlapping groups in social media. Both websites allow users to provide tags to indicate their interests, considered as the features; and users can also subscribe to some interest groups, used as the ground truth as the class labels in this work. In addition, users can follow other users if they share similar interests, forming link information. We compute the number of links for each instance and the distributions are shown in Figure 3. The first observation is that most instances have a few links, while a few instances have an extremely high number of links. These distributions suggest a power law distribution that is typical in social networks. Some statistics of the datasets are shown in Table 4: Flickr has a denser network, indicating by its higher average degree and higher clustering coefficient. Note that "# of Classes" in Table 4 denotes the number of ground-truth classes and the ground-truth classes are only used for the evaluation purpose.

3. the link information $\mathbf{R}$ is very sparse
4. http://www.blogcatalog.com
5. http://www.flickr.com/



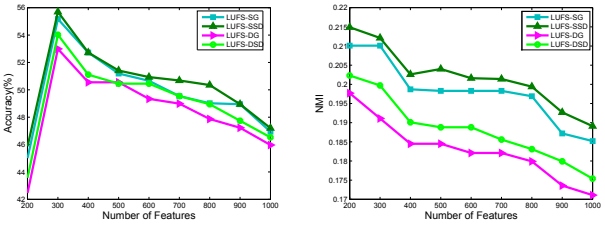
(a) BlogCatalog (b) Flickr

Fig. 3. Link Distribution



(a) Accuracy in Flickr (b) NMI in Flickr

Fig. 4. The Comparison of Variants of LUFS in Flickr



(a) Accuracy in BlogCatalog (b) NMI in BlogCatalog

Fig. 5. The Comparison of Variants of LUFS in BlogCatalog

4.2 Baseline Methods and Metrics

LUFS is compared with the following three representative unsupervised feature selection algorithms: (1) UDFS [38] selects features in batch mode by simultaneously exploiting discriminative information and feature correlation; (2) Laplacian Score [16] evaluates the importance of a feature through its power of locality preservation; (3) SPEC [39] selects features using spectral regression. Note that all these baseline methods have the same time complexity as the proposed framework $o(mn^2)$.

Following the current practice of evaluating unsupervised feature selection, we assess LUFS in terms of clustering performance. The clustering quality is evaluated by two commonly used metrics, *accuracy* and *normalized mutual information* (NMI)[6].

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, and 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.

4.3 Comparison of Four Variants of LUFS

In this section, we compare the quality of selected features by the four variants of LUFS in terms of clustering performance. The purpose of these experiments is two-fold. First we want to investigate the impact of different components of LUFS via its four variants. Second, we have shown the efficiency of these four variants via time complexity analysis and here we want to show their effectiveness. For the parameters in the proposed algorithms, we try different parameter values and report the best performance. More details about parameter analysis will be discussed in the later subsections. The comparison results are demonstrated in Figures 4 and 5 for Flickr and BlogCatalog, respectively. In general, with the increase of the number of selected features, the performance trends to increase first and then decrease. When the number of selected features is smaller, we lose too much information, while the number of selected features is larger, we may introduce noisy features. Algorithms reach their best performance with smaller numbers of selected features in Blogcatalog than in Flickr. For example, most of algorithms achieve their best performance in terms of NMI with 200 selected features in BlogCatalog compared to 700 in Flickr.

We have following observations,

- LUFS-SG always obtains better performance than LUFS-DG, and LUFS-SSD outperforms LUFS-DSD. For example, in BlogCatalog, on average, LUFS-SG and LUFS-SSD obtain 6.54% and 7.33% relative improvement in terms of accuracy comparing to LUFS-DG and LUFS-DSD, respectively. When fixing the components of capturing link information, for the components of capturing attribute-value information, spectral analysis (S) always obtains better performance than discriminative analysis (D).
- LUFS-SSD always outperforms LUFS-SG, and most of the time, LUFS-DSD achieves better performance than LUFS-DG. For instance, in Flickr, the average improvement of LUFS-SSD and LUFS-DSD is 2.59% and 6.78% over LUFS-SG and LUFS-DG with respect to NMI, respectively. When fixing the components of capturing attribute-value information, social dimension regularization is superior to graph regularization in terms of capturing link information. Social dimension regularization captures relations among linked instances as groups while graph regularization considers relations between two linked instances, indicating groups are better than individuals in terms of capturing relations of linked data.

Although LUFS-SSD is not as efficient as other variants as shown in the time complexity analysis subsection, we observe that most of the time, LUFS-SSD obtains the best performance among the four

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

TABLE 5
Clustering Performance with Different Feature Selection Algorithms in Flickr

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

variants of LUFS. For example, comparing to the best performance of other three variants, LUFS-SSD gains 2.59% and 5.92% relative improvement in terms of NMI in Flickr and BlogCatalog, respectively. Therefore in the following experiments, we choose LUFS-SSD for LUFS, indicating that we choose spectral analysis to capture attribute-value information and social dimension regularization to explore link information.

4.4 Quality of Selected Features

In this subsection, we compare the quality of features selected by different algorithms using performance metrics given above. For baseline methods with the parameters, we try different parameter values and report the best performance. 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 datasets. The comparison results of Flickr and BlogCatalog are shown in Tables 5 and 6, 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 [38]. 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 [4], [40], [38]: it is recommended to analyze data features jointly for feature selection. LUFS always obtains the best performance and the relative improvement of LUFS compared to the best performance of baseline method is shown in Table 7. For each number of selected features, we conduct t-test to compare the performance of LUFS and the best performance of baseline methods with significance level 0.01, and these t-test results consistently show that LUFS performs significantly better than baseline methods.

TABLE 7
Improvement of LUFS w.r.t. Performance of Clustering

# Features	Flickr(%)		BlogCatalog(%)	
	Accuracy	NMI	Accuracy	NMI
200	11.03	163.43	11.44	16.67
300	11.74	84.86	7.40	17.38
400	10.66	112.11	6.98	13.25
500	19.82	110.38	4.90	15.65
600	10.38	95.07	6.19	14.22
700	11.52	84.69	7.12	16.75
800	7.29	51.98	7.17	14.33
900	2.32	51.44	4.19	13.22
1000	1.24	33.00	3.65	13.03

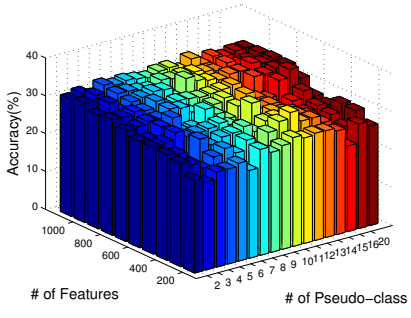
4.5 Parameter Analysis

We will use LUFS-SSD as an example to illustrate parameter selection for LUFS since LUFS-SSD always obtains the best performance. In addition to determining the number of selected features (which remains an open problem [38]), 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.

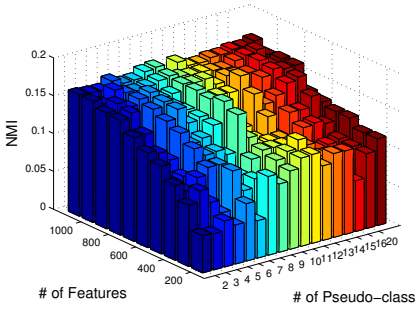
The number of pseudo-class labels c is varied from 2 to 20 with an incremental step of 1 while setting $\{\alpha = 0.1, \beta = 0.1, K = 70\}$. The performance variation w.r.t. c and the number of features is depicted in Figure 6. We note that when c varies from 5 to 11, the clustering performance is not sensitive to c, especially when the number of selected feature is large.

TABLE 6
Clustering Performance with Different Feature Selection Algorithms in BlogCatalog

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

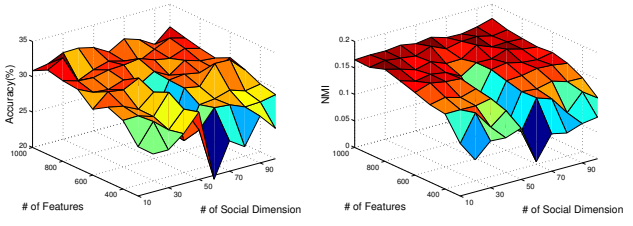


(a) Accuracy



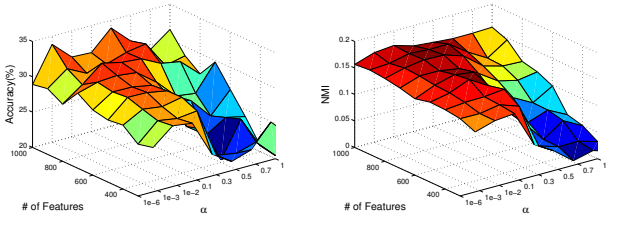
(b) NMI

Fig. 6. Number of Pseudo-class Labels vs Number of Features



(a) Accuracy (b) NMI

Fig. 7. Number of Social Dimension vs Number of Features



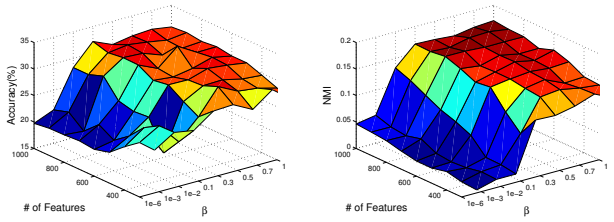
(a) Accuracy (b) NMI

Fig. 8. α 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 7. Most of the time, with the increasing number of social dimensions, the performance first increases, reaches its peak value and degrades. Based on social dimension assumption, labels of instances from the same social dimension are likely to be similar while labels of instances from the different social dimension are likely to be dissimilar. When fewer social dimensions are selected, it is likely that instances with different labels might be mixed in the same social dimension, while more social dimensions are selected, with a high probability, instances with the same label might be divided into several social dimensions. This pattern can be used to determine the optimal number of social dimensions for LUFS.

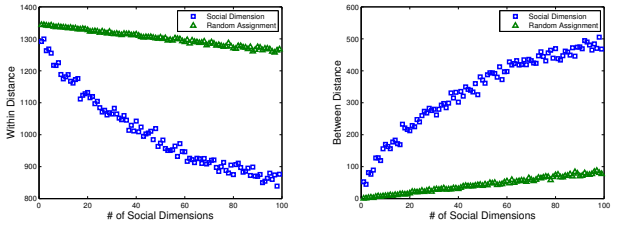
Fixing $c = 9$, $K = 70$ and $\beta = 0.1$, we vary α as $\{1e{-}6, 1e{-}3, 1e{-}2, 0.1, 0.3, 0.5, 0.7, 1\}$. The performance variation w.r.t. α and the number of features is depicted in Figure 8. The performance first increases and most of time, the peak values for both accuracy and NMI are achieved 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 9. 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



(a) Accuracy (b) NMI

Fig. 9. β vs Number of Features



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

Fig. 10. Within and Between Distance Achieved by Social Dimension and Random Assignment in Flickr

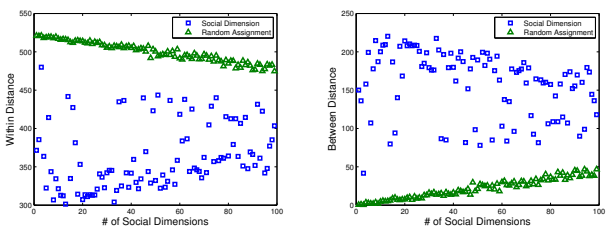
$\ell_{2,1}$-norm for feature selection.

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.

4.6 Probing Further

A key contribution of LUFS to the performance improvement is to exploit link information. Social dimension regularization (SDR) always outperforms graph regularization in terms of capturing link information. SDR employs 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 [26] 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



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

Fig. 11. Within and Between Distance Achieved by Social Dimension and Random Assignment in BlogCatalog

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,

$$D_{within} = Tr(\mathbf{S}_w), \quad D_{between} = Tr(\mathbf{S}_b). \tag{33}$$

With the increase of the number of social dimensions, D_{within} trends to decrease and $D_{between}$ tends to increase in both datasets. 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 10 and Figure 11, respectively.

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, 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 [8], [21] based on the training data being labeled or unlabeled. Recent reviews about supervised and unsupervised feature selection can be found in [32] and [1], respectively.

Supervised feature selection methods [21] can be broadly categorized into the *wrapper* models [9], [19]

and the *filter* models [15], [28]. 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 [7], [14]. 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 [15]. Many researchers paid great attention to developing unsupervised feature selection [37], [16], [5]. Unsupervised feature selection [16], [8], [39] is a less constrained search problem without class labels, depending on clustering quality measures [11], [10], 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 [8] 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 [6], in dimensionality reduction has been widely investigated and applied to feature selection including multi-task feature selection [2], [22], robust joint $\ell_{2,1}$-Norms [27], spectral feature selection [39], discriminative unsupervised feature selection [38]. Through sparsity regularization, feature selection can be embedded in the learning process.

In [13], the authors proposed a supervised feature selection framework FSNet to select features for networked data. It adopts linear regression to fit the content information, and graph regularization to capture the link information. The first attempt to select features on social media data is LinkedFS [31], a semi-supervised algorithm. Various relations (coPost, coFollowing, coFollowed and Following) are extracted following social correlation theories [24]. LinkedFS significantly improves the performance of feature selection by incorporating these relations into feature selection. LinkedFS is substantially different from FSNet - (1) FSNet is supervised while LinkedFS is semi-supervised; (2) LinkedFS extracts four relations from linked data according to social correlation theories, while FSNet utilizes the network directly, which is a special case of these used by LinkedFS; (3) Formulations and optimization algorithms of LinkedFS are different from these of FSNet. Our proposed framework LUFS are distinctively different from both LinkedFS and FSNet. First, LinkedFS and FSNet are semi-supervised and supervised methods, respectively and they use label information, while LUFS is a unsupervised feature selection algorithm. Second, LinkedFS and FSNet exploit 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 study a new problem of selecting feature for linked data in social media. We utilize graph regularization and social dimension regularization to capture the individual and group behaviors of linked instances, separately. In particular, social dimension regularization is based on a recent developed concept of social dimensions from social network analysis to extract relations among linked data as groups and defined to mathematically model these relations. We then propose the concept of pseudo-class labels to guide extracting constraints from link information and attribute-value information, resulting in a new unsupervised feature selection framework, LUFS, for linked social media data. 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.

There are several interesting directions to be investigated. First, in social media networks, the availability of various link formations can lead to networks with relationships of different strengths, which means that weak links and strong links are often mixed together. Since strong links indicate strong correlations among instances, treating all links with a equal weight will increase the level of noise in the learned models and likely lead to degradation in learning performance. One way to further exploit link information is to incorporate tie strength prediction into LUFS.

Second, two characteristics of linked data (i.e., concentrated linkage and autocorrelation) can reduce the effective size of instances for learning. As users in social media are both content producers and content consumers, the quality of social media content can vary drastically. In other words, social media data consists of useful and noisy data instances. Therefore, it makes sense to select instances to help better select relevant features, or vice versa.

Third, another characteristic of social media is that its data comes from a range of multiple sources. For example, tweets in Twitter are always related to external sources such as news from BBC; tags and short text descriptions provide semantic information about photos in Flickr. Having multi-source data can compensate for some problems with social media data. Since data of each source can be noisy, partial, or redundant, selecting relevant sources and using them together can help effective feature selection

Finally, we believe that the concept of pseudo-class labels 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. This work defines a new research problem of feature selection for social media data and

shows its potential and significance, but only touches upon the tip of the iceberg of this fertile research area. Facing challenges arising from learning and mining of social media data, we anticipate many new problems to be researched and a suite of new methods and solutions to be developed.

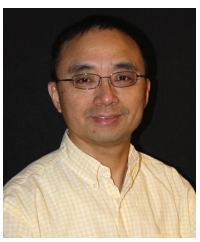
ACKNOWLEDGMENTS

This project is, in part, supported by the National Science Foundation grants #0812551, IIS-1217466,and ONR #N000141410095.

REFERENCES

[1] A. Salem, J. Tang, and H. Liu. Feature selection for clustering: A review. In *Data Clustering: Algorithms and Applications*, *CRC Press*, 2013.
[2] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. *NIPS*, 2007.
[3] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge Univ Pr, 2004.
[4] D. Cai, C. Zhang, and X. He. Unsupervised feature selection for multi-cluster data. In *KDD*, 2010.
[5] C. Constantinopoulos, M. Titsias, and A. Likas. Bayesian feature and model selection for gaussian mixture models. *TPAMI*, 2006.
[6] 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 *ICML*, 2006.
[7] R. Duda, P. Hart, D. Stork, et al. *Pattern classification*, volume 2. wiley New York, 2001.
[8] J. Dy and C. Brodley. Feature selection for unsupervised learning. *JMLR*, 2004.
[9] J. G. Dy and C. E. Brodley. Feature subset selection and order identification for unsupervised learning. In *ICML*, 2000.
[10] J. G. Dy and C. E. Brodley. Visualization and interactive feature selection for unsupervised data. In *KDD*, 2000.
[11] 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. *TPAMI*,2003.
[12] E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. *PNAS*, 2004.
[13] Q. Gu and J. Han. Towards feature selection in network. In *CIKM*, 2011.
[14] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 2002.
[15] M. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *ICML*, 2000.
[16] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. *NIPS*,507–514,2006.
[17] R. Horn and C. Johnson. *Matrix analysis*. Cambridge Univ Pr, 1990.
[18] G. John, R. Kohavi, and K. Pfleger. Irrelevant feature and the subset selection problem. ICML, 1994.
[19] Y. Kim, W. Street, and F. Menczer. Feature selection for unsupervised learning via evolutionary search. In *KDD*, 2000.
[20] H. Liu and H. Motoda. *Computational methods of feature selection*. Chapman & Hall, 2008.
[21] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *TKDE*, 2005.
[22] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient l 2, 1-norm minimization. In *UAI*,2009.
[23] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.
[24] P. Marsden and N. Friedkin. Network studies of social influence. *Sociological Methods and Research*, 22(1):127–151, 1993.
[25] M. McPherson, L. S. Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 2001.
[26] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*,026113, 2004.
[27] F. Nie, H. Huang, X. Cai, and C. Ding. Efficient and robust feature selection via joint l21-norms minimization. NIPS, 1813–1821, 2010.
[28] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *TPAMI*, 2005.
[29] M. Plummer and L. Lovász. *Matching theory*. Access Online via Elsevier, 1986.
[30] V. Roth and T. Lange. Feature selection in clustering problems. *NIPS*,2004.
[31] J. Tang and H. Liu. Feature selection with linked data in social media. In *SDM*, 2012.
[32] J. Tang, A. Salem and H. Liu. Feature selection for classification: A review. In *Data classification: Algorithms and Applications*, *CRC Press*, 2014.
[33] J. Tang and H. Liu. Unsupervised feature selection for linked social media data. In *KDD*, 2012.
[34] L. Tang and H. Liu. Relational learning via latent social dimensions. In *KDD*, 2009.
[35] B. Taskar, P. Abbeel, M. Wong, and D. Koller. Label and link prediction in relational data. In *SRL*, 2003.
[36] X. Wang, L. Tang, H. Gao, and H. Liu. Discovering overlapping groups in social media. In *ICDM*, 2010.
[37] L. Wolf and A. Shashua. Feature selection for unsupervised and supervised inference: the emergence of sparsity in a weighted-based approach. *JMLR*, 2005.
[38] Y. Yang, H. Shen, Z. Ma, Z. Huang, and X. Zhou. L21-norm regularized discriminative feature selection for unsupervised learning. In *IJCAI*, 1589-1594, 2011.
[39] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *ICML*, 2007.
[40] Z. Zhao, L. Wang, and H. Liu. Efficient spectral feature selection with minimum redundancy. In *AAAI*, 2010.

Jiliang Tang is a senior PhD student of Computer Science and Engineering at Arizona State University. He obtained his Master degree in Computer Science and Bachelor degree in Software Engineering at Beijing Institute of Technology in 2008 and 2010, respectively. His research interests are in computing with online trust, mining social media data, social computing, feature selection and data mining. He has published innovatives works in highly ranked journals and top conference proceedings such as ACM TKDD, DMKD, ACM SIGKDD, WWW, WSDM, IJCAI, SDM, ICDM, and CIKM. Updated information can be found at http://www.public.asu.edu/~jtang20.



Dr.Huan Liu is a professor of Computer Science and Engineering at Arizona State University. He obtained his Ph.D. in Computer Science at University of Southern California and B.Eng. in Computer Science and Electrical Engineering at Shanghai JiaoTong University. Before he joined ASU, he worked at Telecom Australia Research Labs and was on the faculty at National University of Singapore. He was recognized for excellence in teaching and research in Computer Science and Engineering at Arizona State University. His research interests are in data mining, machine learning, social computing, and artificial intelligence, investigating problems that arise in many real-world, data-intensive applications with high-dimensional data of disparate forms such as social media. His well-cited publications include books, book chapters, encyclopedia entries as well as conference and journal papers. He serves on journal editorial boards and numerous conference program committees, and is a founding organizer of the International Conference Series on Social Computing, Behavioral-Cultural Modeling, and Prediction (http://sbp.asu.edu/). He is an IEEE Fellow and an ACM Distinguished Scientist. Updated information can be found at http://www.public.asu.edu/~huanliu.