CoSelect: Feature Selection with Instance Selection for Social Media Data

Jiliang Tang* Huan Liu*

Abstract

Feature selection is widely used in preparing high-dimensional data for effective data mining. Attribute-value data in traditional feature selection differs from social media data, although both can be large-scale. Social media data is inherently not independent and identically distributed (*i.i.d.*), but linked. Furthermore, there is a lot of noise. The quality of social media data can vary drastically. These unique properties present challenges as well as opportunities for feature selection. Motivated by these differences, we propose a novel feature selection framework, CoSelect, for social media data. In particular, CoSelect can exploit link information by applying social correlation theories, incorporate instance selection with feature selection, and select relevant instances and features simultaneously. Experimental results on real-world social media datasets demonstrate the effectiveness of our proposed framework and its potential in mining social media data.

1 Introduction

The massive and high dimensional social media data poses new challenges to data mining tasks such as classification and clustering. One conventional and direct approach to handling large-scale and high-dimensional data is feature selection [1, 3]. Feature selection aims to select a subset of relevant features for a compact but more accurate data representation [3, 5].

The vast majority of existing feature selection algorithms work with "flat" data containing uniform entities (or attribute-value data points) that are typically assumed to be independent and identically distributed (*i.i.d.*). However, the nature of social media determines that social media data is different from attribute-value data. Social media data is inherently linked. A typical example of social media data is demonstrated in Figure 1. There are users and posts and note that we use "post" here in a loose way to cover high-dimensional user generated content in social media such as tweets, blogs, photos and videos. A user can have multiple posts (posting relations) and also can follow other users (following relations). Except the conventional attribute-value part (the content of posts), posts are linked via hyperlinks or social context (posting relations and following relations)[1]. Linked social media data is patently not independent and identically distributed, which contradicts one of the most enduring and deeply buried assumptions of traditional machine learning methods [9, 10]. Furthermore, two common characteristics of linked data, i.e., concentrated linkage and autocorrelation, reduce the effective size of instances for learning [9]. That is, given attribute-value data of the same size, there are more irrelevant instances in linked data. The challenges from social media data do not stop here. Users in social media can be both passive content consumers and active content producers, causing the quality of social media data to vary drastically [11]. In other words, social media data consists of useful and noisy data instances, further exacerbating the problem of irrelevant instances.

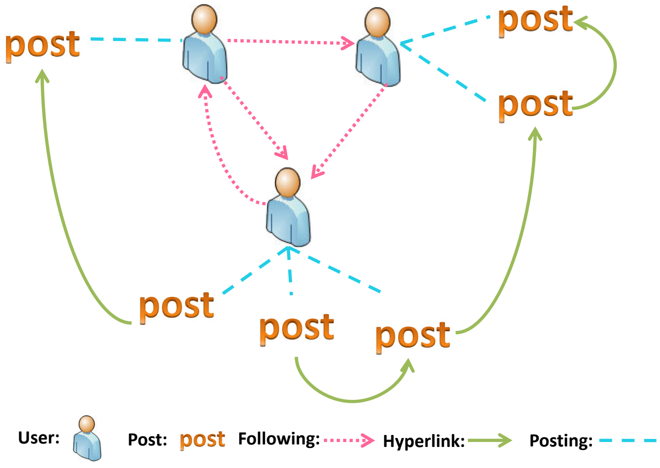


Figure 1: A Typical Example of Social Media Data

However, social media data contains more information than attribute-value data does in terms of links. In general, there are correlations between connected data instances [6, 7]. For example, it is likely that photos

*Computer Science and Engineering, Arizona State University, Tempe, AZ. {jiliang.tang, huan.liu}@asu.edu

[1] For some types of social media data such as photos in Flickr and videos in YouTube, there are no hyperlinks among its data instances, however, social media data consistently has social context.

from the same user have similar topics, or two blogs linked via hyperlinks are more similar in terms of topics. The availability of link information can help advanced research on feature selection for social media data.

In this paper, *we would like to develop a novel framework for performing feature selection on posts (e.g., tweets, blogs, or photos) by considering the unique properties of social media data: availability of link information and existence of irrelevant instances.* We investigate: (1) how to exploit link information; (2) how to select the most relevant instances for feature selection. In our attempt to solve these challenges, we propose a novel feature selection framework, coSelect, for social media data. According to social correlation theories [13, 14], link information can be exploited as topics of linked data instances are more likely to be similar [6]. We incorporate instance selection and feature selection in the same framework to find relevant instances and features simultaneously. Combining instance selection and feature selection enables us to build a novel feature selection framework for linked and noisy social media data: relevant instances help the selection of relevant features, and relevant features in turn help the selection of relevant instances. Our empirical study on two real-world social media datasets demonstrates the effectiveness of CoSelect.

The rest of this paper is organized as follows: our feature selection framework for social media data, CoSelect, with an optimization method and its detailed convergence analysis, is introduced in Section 2; empirical evaluation on datasets from real-world social media websites is presented with discussion in Section 3; the related work is shown in Section 4; and the conclusions and future work are presented in Section 5.

2 A Framework for Social Media Data: coSelect

We first introduce the notations and definitions in our paper. Let $\mathbf{p} = \{p_1, p_2, \dots, p_n\}$ be the set of posts where n is the number of posts and $\mathbf{f} = \{f_1, f_2, \dots, f_m\}$ be the set of features where m is the number of features. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{m\times n}$ where $\mathbf{x}_i$ represents p_i and $\mathbf{x}_i(j)$ is the frequency of f_j used by p_i. Let $\mathbf{c} = \{c_1, c_2, \dots, c_k\}$ denote the class label set where k is the number of classes. $\mathbf{Y} \in \mathbb{R}^{k\times n}$ is the class label indicator matrix where $\mathbf{Y}(j,i) = 1$ if p_i is labeled as c_j, otherwise zero.

Except the conventional representation, social media data is linked. Following [6], p_i and p_j are linked if p_i and p_j are connected with hyperlinks, from the same user, or from two linked users. Let $\mathbf{R} \in \mathbb{R}^{n\times n}$ denote the link information for social media data, and $\mathbf{R}(i,j) = 1$ if p_i and p_j are linked, zero otherwise.

There are two challenges arising from the differences between social media data and attribute-value data: linked and irrelevant instances. In this section, we first present the solutions to these challenges, then with the solutions to the challenges posed by social media data, we propose a novel feature selection framework, coSelect, for social media data, and finally we present an optimization algorithm for coSelect with detailed convergence analysis.

2.1 Exploiting Link Information

Linked social media data is different from attribute-value data due to the correlations between its data instances [6]. Two linked blogs are more likely to have similar topics than two randomly chosen blogs. For example, a blog about "sports" is more likely to connect to other "sports" related blogs. These correlations can be explained by social correlation theories such as homophily [13] and social influence [14]. Homophily suggests that instances with similar topics are more likely to be linked while social influence theory indicates that linked instances are more likely to have similar topics.

We can exploit the correlations suggested by social correlation theories for feature selection. Before going into the details, we first introduce a representative feature selection method based on $\ell_{2,1}$-norm regularization [15] as our basis feature selection algorithm, which selects features across data points with joint sparsity [16].

$$\min_{\mathbf{W}} \; \|\mathbf{W}^\top\mathbf{X} - \mathbf{Y}\|_F^2 + \alpha\|\mathbf{W}\|_{2,1}, \tag{2.1}$$

where $\mathbf{W} \in \mathbb{R}^{m\times k}$ and $\|\mathbf{W}\|_{2,1}$ is the $\ell_{2,1}$-norm of $\mathbf{W}$, which is defined as follows,

$$\|\mathbf{W}\|_{2,1} = \sum_{i=1}^{m}\sqrt{\sum_{j=1}^{k}\mathbf{W}^2(i,j)} = \sum_{i=1}^{m}\|\mathbf{W}(i,:)\|_2 \tag{2.2}$$

it controls the capacity of $\mathbf{W}$ and also ensures that $\mathbf{W}$ is sparse in rows, making it particularly suitable for feature selection. The parameter α is introduced to control the sparsity of $\mathbf{W}$.

Let $T(\mathbf{x}_i) : \mathbb{R}^m \rightarrow \mathbb{R}^k$ be the function to predict the labels of the data instance p_i, i.e., $T(\mathbf{x}_i) = \mathbf{W}^\top\mathbf{x}_i$. As analyzed above, linked data are more likely to have similar labels. The sum of label distances between linked data instances can be calculated as,

$$
\begin{aligned}
&\sum_i \sum_j \mathbf{R}(i,j) \| T(\mathbf{x}_i) - T(\mathbf{x}_j) \|_2^2, \\
&= \sum_i \sum_j \mathbf{R}(i,j) \| \mathbf{W}^\top \mathbf{x}_i - \mathbf{W}^\top \mathbf{x}_j \|_2^2, \\
&= \sum_l \sum_i \sum_j \mathbf{R}(i,j) (\mathbf{w}_l^\top \mathbf{x}_i - \mathbf{w}_l^\top \mathbf{x}_j)^2, \\
&= Tr(\mathbf{W}^\top \mathbf{X} \mathbf{L} \mathbf{X}^\top \mathbf{W}), \qquad (2.3)
\end{aligned}
$$

where $\mathbf{L} = \mathbf{D} - \mathbf{R}$ is the Laplacian matrix, and $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}(i,i) = \sum_j \mathbf{R}(j,i)$. To capture link information, we add a regularization term into Eq. (2.1) to force the labels of linked data instances close to each other by minimizing Eq. (2.3). Therefore the formulation for feature selection exploiting link information can be written as

$$
\min_{\mathbf{W}} \quad \| \mathbf{X}^\top \mathbf{W} - \mathbf{Y} \|_F^2 + \alpha \| \mathbf{W} \|_{2,1} + \gamma Tr(\mathbf{W}^\top \mathbf{X} \mathbf{L} \mathbf{X}^\top \mathbf{W}), \qquad (2.4)
$$

where γ is introduced to control the contribution from link information.

2.2 Incorporating Instance Selection Social media produces massive user generated content and the quality of social media data varies drastically from excellent content to abuse and spam [11], resulting in useful and irrelevant instances mixed together. Furthermore, the linked property of social media data determines the existence of irrelevant instances [9]. In this subsection, we introduce the technical details about how to select relevant instances for feature selection.

Let $\mathbf{B} = \mathbf{X}^\top \mathbf{W} - \mathbf{Y}^\top - \mathbf{\Xi}$ be the residual matrix where $\mathbf{\Xi}$ is a random error matrix, usually assumed to be multi-dimensional normal distribution [17]. The residual matrix $\mathbf{B}$ is a strong indicator of anomalies in a dataset [18, 17]. Each row of $\mathbf{B}$ corresponds to a data instance, and a large norm of $\mathbf{B}(i,:)$ indicates a significant deviation of the i-th data instance, more likely to be an irrelevant instance. Therefore the residual matrix $\mathbf{B}$ can be used to realize instance selection and feature selection with instance selection can be formulated as:

$$
\min_{\mathbf{W},\mathbf{B}} \quad \| \mathbf{W}^\top \mathbf{X} - \mathbf{B}^\top - \mathbf{Y} \|_F^2 + \alpha \| \mathbf{W} \|_{2,1} + \beta \| \mathbf{B} \|_{2,1}, \qquad (2.5)
$$

where the parameter β is introduced to control the sparsity of $\mathbf{B}$.

In Eq. (2.5), feature selection is achieved via $\ell_{2,1}$-norm on $\mathbf{W}$ while the purpose of the $\ell_{2,1}$-norm on $\mathbf{B}$ is two-fold. One is that $\ell_{2,1}$-norm of a matrix ensures that $\mathbf{B}$ is sparse in rows, making it particularly suitable for instance selection; the other is to avoid the trivial estimate $\mathbf{B} = \mathbf{Y}^\top$ and get a meaningful estimate of $\mathbf{W}$.

2.3 The CoSelect Framework With our solutions to these two challenges posed by social media data, we are ready to introduce the CoSelect framework. Considering both exploiting link information (Eq. (2.4)) and incorporating instance selection (Eq. (2.5)), CoSelect is to solve the following optimization problem,

$$
\begin{aligned}
\min_{\mathbf{W},\mathbf{B}} \ & \| \mathbf{W}^\top \mathbf{X} - \mathbf{B}^\top - \mathbf{Y} \|_F^2 + \alpha \| \mathbf{W} \|_{2,1} \\
& + \beta \| \mathbf{B} \|_{2,1} + \gamma Tr(\mathbf{W}^\top \mathbf{X} \mathbf{L} \mathbf{X}^\top \mathbf{W}). \qquad (2.6)
\end{aligned}
$$

We use $\mathcal{J}(\mathbf{W},\mathbf{B})$ to denote the objective function in Eq. (2.6), where the first term is used to fit the label information, the second and third terms use $\ell_{2,1}$ on $\mathbf{W}$ and $\mathbf{B}$ to achieve feature selection and instance selection respectively, and the fourth term is used to exploit link information. It is easy to verify that $\mathcal{J}(\mathbf{W},\mathbf{B})$ is jointly convex with $\mathbf{W}$ and $\mathbf{B}$, implying that a global optimal solution is guaranteed for the problem in Eq. (2.6). However, it is difficult to optimize two variables simultaneously. Thus we adopt an alternating optimization to solve this problem, which works well for a number of practical optimization problems [19]. Under this scheme, we update $\mathbf{W}$ and $\mathbf{B}$ in an alternating manner.

2.3.1 Given B, Computing W: Given $\mathbf{B}$, the optimal $\mathbf{W}$ can be obtained by minimizing the following problem,

$$
\begin{aligned}
\min_{\mathbf{W}} \ & \mathcal{J}(\mathbf{W}) = \| \mathbf{W}^\top \mathbf{X} - \mathbf{C} \|_F^2 + \\
& \alpha \| \mathbf{W} \|_{2,1} + \gamma Tr(\mathbf{W}^\top \mathbf{X} \mathbf{L} \mathbf{X}^\top \mathbf{W}), \qquad (2.7)
\end{aligned}
$$

where $\mathbf{C} = \mathbf{B}^\top + \mathbf{Y}$ is a constant matrix. We develop the following theorem to obtain the update rule for $\mathbf{W}$ in each iteration.

THEOREM 2.1. *When fixing* $\mathbf{B}$, $\mathbf{W}$ *can be updated by,*

$$
\mathbf{W} \leftarrow (\mathbf{X}\mathbf{X}^\top + \gamma \mathbf{X}\mathbf{L}\mathbf{X}^\top + \alpha \mathbf{D}_W)^{-1} \mathbf{X}\mathbf{C}^\top, \qquad (2.8)
$$

where $\mathbf{D}_W$ *is a diagonal matrix with the* i*-th diagonal element as:* $\mathbf{D}_W(i,i) = \frac{1}{2\|\mathbf{W}(i,:)\|_2}$.

Proof. The Lagrangian function of the problem in Eq. (2.7) is:

$$
\begin{aligned}
\mathcal{L}_W &= Tr(\mathbf{W}^\top \mathbf{X}\mathbf{X}^\top \mathbf{W} - 2\mathbf{W}^\top \mathbf{X}\mathbf{C}^\top) \\
&+ \alpha \| \mathbf{W} \|_{2,1} + \gamma Tr(\mathbf{W}^\top \mathbf{X} \mathbf{L} \mathbf{X}^\top \mathbf{W}). \qquad (2.9)
\end{aligned}
$$

We take the derivative of $\mathcal{L}_W$,

$$\frac{\partial \mathcal{L}_W}{\partial \mathbf{W}} = 2\mathbf{X}\mathbf{X}^\top\mathbf{W} - 2\mathbf{X}\mathbf{C}^\top + 2\alpha\mathbf{D}_W\mathbf{W} + 2\gamma\mathbf{X}\mathbf{L}\mathbf{X}^\top\mathbf{W}.$$

Matrices $\mathbf{X}\mathbf{X}^\top$ and $\mathbf{X}\mathbf{L}\mathbf{X}^\top$ are semi-positive definite matrices and therefore $\mathbf{X}\mathbf{X}^\top + \gamma\mathbf{X}\mathbf{L}\mathbf{X}^\top + \alpha\mathbf{D}_W$ is a positive definite matrix. Setting the derivative to zero, we can obtain the update rule in Eq. (2.10),

$$\mathbf{W} = (\mathbf{X}\mathbf{X}^\top + \gamma\mathbf{X}\mathbf{L}\mathbf{X}^\top + \alpha\mathbf{D}_W)^{-1}\mathbf{X}\mathbf{C}^\top, \tag{2.10}$$

which completes the proof.

2.3.2 Given W, Computing B: Given $\mathbf{W}$, the optimal $\mathbf{B}$ can be obtained by minimizing the following problem,

$$\min_{\mathbf{B}} \quad \mathcal{J}(\mathbf{B}) = \|\mathbf{H} - \mathbf{B}^\top\|_F^2 + \beta\|\mathbf{B}\|_{2,1}, \tag{2.11}$$

where $\mathbf{H} = \mathbf{W}^\top\mathbf{X} - \mathbf{Y}$.

Similarly, we develop the following theorem to obtain the update rule for $\mathbf{B}$.

Theorem 2.2. *When fixing $\mathbf{W}$, $\mathbf{B}$ can be updated by,*

$$\mathbf{B} \leftarrow (\mathbf{I}_n + \beta\mathbf{D}_B)^{-1}\mathbf{H}^\top, \tag{2.12}$$

where $\mathbf{D}_B$ is a diagonal matrix with the i-th diagonal element as: $\mathbf{D}_B(i,i) = \frac{1}{2\|\mathbf{B}(i,:)\|_2}$.

Proof. The proof process is similar to that for Theorem 2.1. To save space, we ignore the detailed proof.

Based on Theorems 2.1 and 2.2, we develop CoSelect with its details in Algorithm 1: by setting $\mathbf{C} = \mathbf{Y}$ in Eq. (2.10), $\mathbf{W}$ is initialized as shown in line 3, in order to speed up the algorithm's convergence (more to follow in the experiment section). From line 5 to line 10, $\mathbf{B}$ and $\mathbf{W}$ are updated alternatingly according to Theorems 2.1 and 2.2. After the algorithm terminates, we select the most relevant features and instances according to $\mathbf{W}$ and $\mathbf{B}$, respectively. The larger the norm of $\|\mathbf{W}(i,:)\|_2$, the more relevant the i-th feature is; while the larger the norm of $\|\mathbf{B}(j,:)\|_2$, the more irrelevant the j-th instance is, as shown in lines 12 and 13. We begin the convergence study with the following lemma.

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

$$\sum_i \|\mathbf{b}_{t+1}^i\|_2 - \sum_i \frac{\|\mathbf{b}_{t+1}^i\|_2}{2\|\mathbf{b}_t^i\|_2} \le \sum_i \|\mathbf{b}_t^i\|_2 - \sum_i \frac{\|\mathbf{b}_t^i\|_2^2}{2\|\mathbf{b}_t^i\|_2} \tag{2.13}$$

Algorithm 1 CoSelect

```
Input: $\{\mathbf{X}, \mathbf{Y}, \mathbf{R}, \alpha, \beta, \gamma, k_f, k_i\}$
Output: $k_f$ most relevant features and $k_i$ most relevant instances
1: Construct the Laplacian matrix $\mathbf{L}$;
2: Initialize $\mathbf{D}_W$ and $\mathbf{D}_B$ as identity matrices;
3: Initialize $\mathbf{W} = (\mathbf{X}\mathbf{X}^\top + \gamma\mathbf{X}\mathbf{L}\mathbf{X}^\top + \alpha\mathbf{D}_W)^{-1}\mathbf{X}\mathbf{Y}^\top$;
4: while Not convergent do
5:     Set $\mathbf{H} = \mathbf{W}^\top\mathbf{X} - \mathbf{Y}$;
6:     Update $\mathbf{B}$ as: $\mathbf{B} \leftarrow (\mathbf{I}_n + \beta\mathbf{D}_B)^{-1}\mathbf{H}^\top$;
7:     Update the diagonal matrix $\mathbf{D}_B$, where the $i$-th diagonal element is $\frac{1}{2\|\mathbf{B}(i,:)\|_2}$;
8:     Set $\mathbf{C} = \mathbf{B}^\top + \mathbf{Y}$;
9:     Update $\mathbf{W}$ as: $\mathbf{W} \leftarrow (\mathbf{X}\mathbf{X}^\top + \gamma\mathbf{X}\mathbf{L}\mathbf{X}^\top + \alpha\mathbf{D}_W)^{-1}\mathbf{X}\mathbf{C}^\top$;
10:    Update the diagonal matrix $\mathbf{D}_W$, where the $i$-th diagonal element is $\frac{1}{2\|\mathbf{W}(i,:)\|_2}$;
11: end while
12: Sort each feature according to $\|\mathbf{W}(i,:)\|_2$ in descending order and select the top-$k_f$ ranked ones;
13: Sort each instances according to $\|\mathbf{B}(j,:)\|_2$ in ascending order and select the top-$k_i$ ranked ones.
```

Proof. The proof process is similar to that in [20, 6], and we ignore the detailed proof to save space.

With Lemma 1, we develop the following theorem regarding the convergence of Algorithm 1.

Theorem 2.3. *Algorithm 1 monotonically decreases the objective function value of $\mathcal{J}(\mathbf{W}, \mathbf{B})$ in Eq. (2.6) thus Algorithm 1 converges.*

Proof. We divide the proof process into two steps: (1) the update rule in Theorem 2.2 monotonically decreases $\mathcal{J}(\mathbf{B})$, i.e., $\mathcal{J}(\mathbf{W}_t, \mathbf{B}_t) \ge \mathcal{J}(\mathbf{W}_t, \mathbf{B}_{t+1})$, and (2) the update rule in Theorem 2.1 monotonically decreases $\mathcal{J}(\mathbf{W})$, i.e., $\mathcal{J}(\mathbf{W}_t, \mathbf{B}_{t+1}) \ge \mathcal{J}(\mathbf{W}_{t+1}, \mathbf{B}_{t+1})$.

It can be easily verified that $\mathbf{B}_{t+1}$ is the solution to the following problem,

$$\mathbf{B}_{t+1} = \min_{\mathbf{B}} Tr(\mathbf{B}^\top(\mathbf{I}_n + \beta\mathbf{D}_B)\mathbf{B} - 2\mathbf{B}^\top\mathbf{H}^\top),$$

which indicates that,

$$\begin{aligned} &Tr(\mathbf{B}_{t+1}^\top(\mathbf{I}_n + \beta\mathbf{D}_B)\mathbf{B}_{t+1} - 2\mathbf{B}_{t+1}^\top\mathbf{H}^\top) \\ &\le Tr(\mathbf{B}_t^\top(\mathbf{I}_n + \beta\mathbf{D}_B)\mathbf{B}_t - 2\mathbf{B}_t^\top\mathbf{H}^\top). \end{aligned}$$

Then we have the following inequality,

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

Meanwhile, according to Lemma 1, we have,

$$\|\mathbf{H} - \mathbf{B}_{t+1}^\top\|_F^2 + \beta\|\mathbf{B}_{t+1}\|_{2,1} \leq \|\mathbf{H} - \mathbf{B}_t^\top\|_F^2 + \beta\|\mathbf{B}_t\|_{2,1},$$

which indicates that $\mathcal{J}(\mathbf{W}_t, \mathbf{B}_t) \geq \mathcal{J}(\mathbf{W}_t, \mathbf{B}_{t+1})$. Similarly, we can verify that $\mathcal{J}(\mathbf{W}_t, \mathbf{B}_{t+1}) \geq \mathcal{J}(\mathbf{W}_{t+1}, \mathbf{B}_{t+1})$. Thus we can get the following inequality chain,

$$
\begin{aligned}
&\mathcal{J}(\mathbf{W}_0, \mathbf{B}_0) \geq \mathcal{J}(\mathbf{W}_0, \mathbf{B}_1) \\
&\quad \geq \mathcal{J}(\mathbf{W}_1, \mathbf{B}_1) \geq \mathcal{J}(\mathbf{W}_1, \mathbf{B}_2) \ldots .
\end{aligned}
$$

Algorithm 1 monotonically decreases the objective function value. Thus it converges to an optimal solution for Eq. (2.6) since the objective function in Eq. (2.6) is jointly convex, which completes the proof.

Time Complexity: The most time consuming operations of Algorithm 1 are to obtain the inverses of the matrices $\mathbf{I}_n + \beta\mathbf{D}_B$ and $(\mathbf{X}\mathbf{X}^\top + \gamma\mathbf{X}\mathbf{L}\mathbf{X}^\top + \alpha\mathbf{D}_W)$.

- $\mathbf{I}_n + \beta\mathbf{D}_B$ is a diagonal matrix, hence, $(\mathbf{I}_n + \beta\mathbf{D}_B)^{-1}$ is also diagonal and its i-th diagonal element correspond to the reciprocal of i-th diagonal element of $\mathbf{I}_n + \beta\mathbf{D}_B$, which needs $O(n)$ operations.
- The terms $\mathbf{X}\mathbf{X}^\top$ and $\mathbf{X}\mathbf{L}\mathbf{X}^\top$ in $(\mathbf{X}\mathbf{X}^\top + \gamma\mathbf{X}\mathbf{L}\mathbf{X}^\top + \alpha\mathbf{D}_W)$ can be computed efficiently since $\mathbf{X}$ is sparse. It costs $O(\max(m,n)n)$. They are not changed during iterations, hence we only need to calculate them once. $\mathbf{W} = (\mathbf{X}\mathbf{X}^\top + \gamma\mathbf{X}\mathbf{L}\mathbf{X}^\top + \alpha\mathbf{D}_W)^{-1}\mathbf{X}\mathbf{C}^\top$ can be efficiently obtained through solving the linear equation $(\mathbf{X}\mathbf{X}^\top + \gamma\mathbf{X}\mathbf{L}\mathbf{X}^\top + \alpha\mathbf{D}_W)\mathbf{W} = \mathbf{X}\mathbf{C}^\top$, which needs $O(m^2k)$.

In summary, the total time complexity of Algorithm 1 is $\#iterations * O(m^2k) + O(\max(m,n)n)$

3 Experiments

In this section, we conduct experiments to verify the effectiveness of the proposed framework, CoSelect, on datasets from real-world social media websites by comparing it with the state-of-the-art feature selection methods *with and without* instance selection. After introducing datasets, we discuss experiment setting with a convergence study, and present a comprehensive comparative study of CoSelect followed by a discussion on parameter selection.

Table 1: Statistics of the Datasets

Datasets	BlogCatalog	Digg
# Posts	4,142	1,943
# Features	4,548	3,289
# Classes	8	2
# Links	10,242	8,051

3.1 Social Media Data Two datasets from real-world social media websites, BlogCatalog and Digg, are collected for evaluation.

BlogCatalog: BlogCatalog[2] is a blog directory where users can register their blogs under predefined categories, which are used as class labels of blogs in our work. Blogs are linked via their hyperlinks and social context. This dataset is obtained from [21].

Digg: Digg[3] is a popular social news aggregator that allows users to submit, digg, and comment on stories. The topics of stories are considered as the class labels. The hyperlinks between stories are unavailable and stories are linked via social context. This dataset is a subset of the dataset used in [22].

The posts are preprocessed for stop-word removal and stemming and TFIDF is used to remove obviously irrelevant features. Note that terms in posts are considered as features in both data sets. Some statistics of these datasets are shown in Table 1.

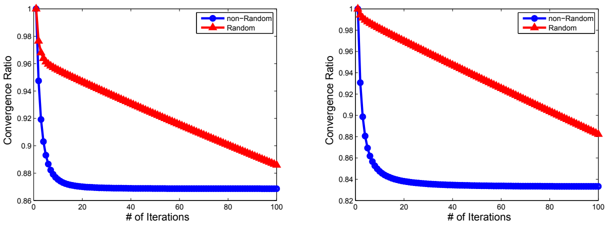
3.2 Experiment Setting and Baseline Methods

For both datasets, we randomly split data into training data (40%) and testing data (60%). Following [20, 23], feature quality is assessed via classification performance. Linear SVM [25] is used for classification given its overall good performance. How to automatically determine the optimal number of selected features is still an open problem [6], thus we vary the percentages of selected features from 10% to 100% with an incremental step of 10%.

Two representative feature selection algorithms are chosen as baseline methods: FisherScore [26](FS) and Joint $\ell_{2,1}$-Norms [20](RFS). Previous study showed that FisherScore is a state-of-the-art supervised feature selection algorithm [27] while RFS applies joint $\ell_{2,1}$-norm minimization on both loss function and regularization,

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

[3] http://www.digg.com



(a) BlogCatalog

(b) Digg

Figure 2: Convergence Ratio of CoSelect with Different Initializing Methods. Note that in this figure, “Random” denotes initializing $\mathbf{W}$ randomly while “non-Random” denotes initializing $\mathbf{W}$ as shown in line 3 of Algorithm 1.

which is robust to outliers [20]. *We did not compare CoSelect with LinkedFS in [6] as LinkedFS utilizes unlabeled data during training.* As a common practice, the parameters in feature selection algorithms and Linear SVM are tuned via cross-validation.

3.3 Convergence Study Before proceeding to obtain classification results, we investigate the convergence of CoSelect with two different initializations of $\mathbf{W}$, random vs. non-random initialization; *the latter initialization is shown in line 3 of Algorithm 1.* The results are depicted in Figure 2. We observe that: (1) CoSelect with non-random initialization of $\mathbf{W}$ converges much faster than that with random initialization; and (2) CoSelect with non-random initialization converges rapidly, usually in fewer than 10 iterations. In our following experiments, we adopt the non-random initialization of $\mathbf{W}$ in pursuit of efficiency.

3.4 Comparing CoSelect with Feature Selection Algorithms with and without Instance Selection In this set of experiments, we evaluate CoSelect in two scenarios: (1) without applying instance selection, we observe how different feature selection algorithms fare, and (2) with instance selection, i.e., applying instance selection with state-of-the-art algorithms first, we examine how these feature selection algorithms compare with CoSelect. When using 100% of instances, we select $x\%$ of features whose classification performance is shown in Table 2, where x varies from 10 to 100 with an increment of 10. Note that all parameters are determined through cross-validation. For CoSelect, we set $\{\alpha = 0.1, \beta = 0.01, \lambda = 0.3\}$ for BlogCatalog and $\{\alpha = 0.1, \beta = 0.1, \lambda = 0.3\}$ for Digg.

As the percentage of selected features increases, the performance first improves, achieves its peak value, and then degrades. In short, feature selection helps. For example, CoSelect gains 16.24% and 8.85% relative improvement with 20% of features in BlogCatalog and Digg, respectively, compared to that with 100% of features. It is clear that the number of features can be significantly reduced for social media data without compromising performance. We note that RFS consistently outperforms FS. RFS selects features in batch mode and considers feature correlation. It is consistent with what was suggested in [6, 23] that it is better to analyze features jointly for feature selection. CoSelect consistently outperforms RFS (and FS), especially when the numbers of selected features are small. There are two main reasons: first, CoSelect exploits link information among instances; and second, CoSelect selects the most relevant features and instances simultaneously.

Next we run FS and RFS after applying an instance selection algorithm to select instances, and we choose DROP3 in our work since it is proven to be one of the state-of-the-art instance selection algorithms [28]. We vary the number of selected instances from 10% to 100% with an incremental step of 10%, and then for each percentage of selected instances, we repeat the experiment by selecting $x\%$ features where x varies from 10 to 100 with an increment of 10. Hence, for each $y\%$ of selected instances, there are 10 accuracy rates correspondingly, and we report the best performance with the corresponding selected features and the results are shown in Table 3.

A simple combination of feature selection, i.e., FS or RFS, and instance selection, i.e., DROP3, results in limited or no performance improvement for social media data. For example, FS only gains less than 1% improvement in BlogCatalog and no improvement in Digg. As we know, social media data is high-dimensional and noisy. Therefore high-dimensional data can make instance selection difficult and noisy instances could confuse feature selection. CoSelect obtains significant improvement when selecting features and instances simultaneously: on average, CoSelect obtains 9.76% and 4.06% relative improvement in BlogCatalog and Digg, respectively.

We note that CoSelect achieves its best performance with fewer instances and features, demonstrating that CoSelect is more likely to select relevant instances and features. CoSelect allows feature selection and instance selection to reinforce each other: relevant instances can guide the selection of relevant features, and relevant features in turn help select relevant instances.

3.5 A Comprehensive Study of Co-Selecting Features and Instances In this subsection, we systematically investigate the performance of CoSelect by

Table 2: Classification Accuracy of Different Feature Selection Algorithms with 100% Instances

Features	BlogCatalog			Digg		
	FS	RFS	CoSelect	FS	RFS	CoSelect
10%	48.45%	48.98%	51.78%	75.70%	74.51%	79.74%
20%	48.65%	50.22%	52.35%	86.60%	86.54%	87.98%
30%	50.90%	51.67%	52.57%	85.06%	86.61%	88.64%
40%	51.12%	51.79%	52.57%	84.46%	84.81%	87.15%
50%	51.35%	51.57%	51.79%	84.55%	85.06%	86.23%
60%	51.35%	51.80%	51.93%	83.38%	84.89%	86.06%
70%	51.57%	51.63%	51.65%	83.18%	84.03%	84.78%
80%	50.67%	50.77%	50.80%	83.78%	83.78%	83.78%
90%	49.33%	49.42%	49.55%	82.32%	82.32%	82.32%
100%	48.21%	48.21%	48.21%	81.46%	81.46%	81.46%

varying numbers of both selected features and selected instances from 10% to 100% with an incremental step of 10%: i.e., a total of 100 experiments for each dataset. We only show the results for BlogCatalog in Table 4 since we have similar observations in Digg. Note that the first row represents the percentages of features and the first column denotes the percentages of instances.

The highlighted numbers are the best performance of CoSelect for each percentage of selected features. It is consistently better than that with 100% instances. The boldfaced numbers signify the cases with the smallest numbers of features and instances whose performance exceeds that with 100% features and 100% instances. We can see that using a very small portion of the whole datasets can obtain comparable or better performance than using the whole dataset. For example, for the BlogCatalog data, the performance with 40% instances and 10% of features is better than that with 100% instances and 100% of features, i.e., 51.10% vs 48.21%. In other words, CoSelect can obtain better performance with only 4% of the whole dataset.

In order to verify the effect of instance selection with respect to feature selection, given a percentage of selected features, we perform another comparison for CoSelect to use selected instances or the full set of training data in classification. We observe that when selecting relevant features and instances together, CoSelect gains up to 10.36% and 4.09% relative improvement for BlogCatalog and Digg, respectively. The above observations demonstrate that CoSelect can select relevant features and instances simultaneously by integrating instance and feature selection in the same framework.

4 Related Work

Feature selection methods fall into three categories, i.e., the filter model, the wrapper model and the embedded model [29, 5]. The filter model relies on general characteristics of the data to evaluate and select feature subsets without involving any mining algorithm [30, 31, 32, 24]. The wrapper model requires one predetermined mining algorithm and uses its performance as the evaluation criterion. It searches for features better suited to the mining algorithm aiming to improve mining performance [26, 34, 35]. For the embedded model, the procedure of feature selection is embedded directly in the training process [36, 20, 15, 6, 4].

Sparsity regularization such as $\ell_{2,1}$ of a matrix is applied to feature selection in recent years. In [16], the $\ell_{2,1}$ of the matrix is introduced as a rotational invariant ℓ_1 norm. A similar model for $\ell_{2,1}$-norm regularization is proposed to select features for multi-task learning [37, 15]. In [23], Zhao et al. introduces a spectral feature selection algorithm based on a sparse multi-output regression with a $\ell_{2,1}$ norm constraint to select relevant features while removing redundancy. Nie et al. [20] apply $\ell_{2,1}$-norm minimization to both loss function and regularization. A loss function with $\ell_{2,1}$-norm is robust to outliers while $\ell_{2,1}$-norm regularization selects features across all instances with joint sparsity. The increasing popularity of social media produces massive linked social media data. In [6], LinkedFS is proposed to take advantage of social context for feature selection in a semi-supervised manner. In [4], LUFS is introduced to exploit link information presented in social media data for unsupervised feature selection. However, both LinkedFS and LUFS utilize unlabeled data when training and cannot distinguish relevant instances from irrelevant instances for effective learning.

In [40], a coSelection framework is proposed for unsupervised rare category analysis. It can co-select examples from the rare category and features relevant to identify the rare category. However, this coSelection framework is different from our proposed coSelect framework. First, coSelection in [40] is only for attribute-value data while our proposed framework can take into account

Table 3: Classification Accuracy of Different Feature Selection Algorithms with Different Numbers of Selected Instances

Instances	BlogCatalog			Digg		
	FS(#)	RFS(#)	CoSelect(#)	FS(#)	RFS (#)	CoSelect(#)
10%	26.10%(70%)	28.02%(70%)	40.57%(40%)	56.74%(50%)	56.98%(60%)	64.37%(40%)
20%	30.41%(70%)	31.06%(60%)	44.38%(40%)	60.55%(50%)	61.63%(50%)	68.06%(30%)
30%	34.65%(70%)	34.54%(70%)	48.19%(50%)	70.37%(50%)	69.47%(50%)	80.99%(40%)
40%	36.39%(70%)	38.02%(70%)	52.00%(20%)	75.87%(50%)	75.90%(40%)	87.21%(50%)
50%	38.01%(70%)	38.02%(60%)	51.55%(50%)	75.44%(40%)	78.54%(40%)	86.78%(40%)
60%	43.54%(60%)	43.55%(60%)	55.36%(20%)	80.67%(40%)	81.53%(40%)	87.46%(40%)
70%	47.07%(60%)	48.02%(60%)	57.70%(40%)	82.75%(40%)	82.66%(30%)	88.15%(30%)
80%	51.97%(60%)	52.36%(60%)	56.04%(40%)	83.35%(40%)	84.46%(30%)	92.21%(30%)
90%	52.07%(70%)	53.52%(60%)	53.45%(30%)	83.92%(40%)	85.65%(30%)	89.09%(30%)
100%	51.57%(70%)	51.80%(60%)	52.57%(40%)	86.60%(20%)	86.61%(30%)	88.64%(30%)

linked data. Second, the tasks of these two frameworks are different. coSelection aims to identify the rare categories and their relevant features, including rare category selection and feature selection, while our work focuses on feature selection aided by instance selection for efficient and effective learning. Instance selection is introduced to address the noisiness of social media data. Third, coSelection in [40] is unsupervised while ours is supervised. Thus their formulations and optimization methods are different from ours.

5 Conclusions

Having observed the unique characteristics of social media data, we propose a novel feature selection framework, CoSelect, for social media data. As we know, social media data is linked, noisy, and can contain irrelevant instances. Link information suggests that topics of linked instances may be similar. Instance selection is incorporated into feature selection in order to select relevant instances while selecting features. Experimental results on two real-world social media datasets show that the proposed framework (CoSelect) can effectively handle linked data to select features as well as instances. By performing feature selection and instance selection in a co-selection scheme, CoSelect can greatly reduce data (e.g., only 4% of the original dataset) without performance deterioration.

In online social networks, it is easy to establish connections which can lead to networks with heterogeneous relationship strengths [41]. Thus one future direction is to incorporate tie strength estimation and prediction into CoSelect to further exploit link information. Except link information, social media data is always related to other sources. In the future, we would like to investigate how to integrate multiple sources about social media data for feature selection

Acknowledgments

We thank the anonymous reviewers for their useful comments. The work is, in part, supported by NSF (#IIS-1217466).

References

[1] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Machine learning*, 2002.

[2] Bezdek, J. and Hathaway, R., “Some notes on alternating optimization”, in *AFSS*, 2002.

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

[4] J. Tang, and H. Liu, “Unsupervised feature selection for linked social media data,” in *KDD*, 2012.

[5] S. Alelyani, J. Tang, and H. Liu, “Feature Selection for Clustering: A Review,” in *Data Clustering: Algorithms and Applications, Editor: Charu Aggarwal and Chandan Reddy, CRC Press*, 2013.

[6] J. Tang and H. Liu, “Feature selection with linked data in social media,” in *SDM*, 2012.

[7] J. Tang, H. Gao, X. Hu and H. Liu, “Exploiting Homophily Effect for Trust Prediction,” in *WSDM*,2013.

[8] Y. Hu, A. John, F. Wang,and S. Kambhampati. ET-LDA: Joint Topic Modeling for Aligning Events and their Twitter Feedback. In *AAAI*, 2012.

[9] D. Jensen and J. Neville, “Linkage and autocorrelation cause feature selection bias in relational learning,” in *ICML*, 2002.

[10] B. Taskar, P. Abbeel, M. Wong, and D. Koller, “Label and link prediction in relational data,” in *Proceedings of the IJCAI Workshop on Learning Statistical Models from Relational Data*, 2003.

[11] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, “Finding high-quality content in social media,” in *WSDM*, 2008.

[12] H. Liu and H. Motoda, “On issues of instance selection,” *DMKD*, 2002.

Table 4: Performance of Feature Selection with Instance Selection in BlogCatalog. Note that the first row represents the percentages of features while the first column denotes the percentages of instances.

	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
10%	38.55%	39.22%	40.57%	40.57%	39.44%	39.44%	39.44%	39.44%	39.67%	39.00%
20%	42.13%	43.26%	43.70%	44.38%	41.91%	41.69%	40.79%	39.44%	38.55%	35.86%
30%	46.39%	47.52%	47.96%	47.96%	48.19%	46.84%	46.84%	46.17%	45.27%	42.58%
40%	**51.10%**	52.00%	49.31%	51.10%	50.21%	49.98%	48.41%	47.29%	45.05%	42.58%
50%	51.33%	51.33%	51.33%	51.55%	51.55%	50.88%	50.88%	50.21%	49.53%	47.52%
60%	53.79%	55.36%	53.35%	53.35%	52.67%	52.90%	52.90%	53.12%	50.65%	48.86%
70%	54.91%	56.04%	56.48%	57.61%	57.16%	56.71%	56.48%	55.14%	54.02%	52.45%
80%	54.02%	54.24%	55.14%	56.04%	55.59%	55.81%	55.14%	54.02%	54.24%	52.67%
90%	52.00%	53.00%	53.45%	52.67%	51.33%	50.43%	50.21%	49.53%	49.53%	47.74%
100%	51.78%	52.35%	52.57%	52.57%	51.79%	51.93%	51.65%	50.80%	49.55%	48.21%

[13] M. McPherson, L. S. Lovin, and J. M. Cook, “Birds of a feather: Homophily in social networks,” *Annual Review of Sociology*, 2001.

[14] P. Marsden and N. Friedkin, “Network studies of social influence,” *Sociological Methods and Research*, 1993.

[15] J. Liu, S. Ji, and J. Ye, “Multi-task feature learning via efficient l 2, 1-norm minimization,” in *UAI*, 2009.

[16] 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.

[17] Y. She and A. Owen, “Outlier detection using nonconvex penalized regression,” *Journal of the American Statistical Association*, 2011.

[18] H. Tong and C. Lin, “Non-negative residual matrix factorization with application to graph anomaly detection.” SDM, 2011.

[19] J. Bezdek and R. Hathaway, “Some notes on alternating optimization,” *Advances in Soft Computing*, 002.

[20] F. Nie, H. Huang, X. Cai, and C. Ding, “Efficient and robust feature selection via joint l21-norms minimization.” in *NIPS*, 2010.

[21] X. Wang, L. Tang, H. Gao, and H. Liu, “Discovering overlapping groups in social media,” in *ICDM*,2010.

[22] Y. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher, “Metafac: community discovery via relational hypergraph factorization,” in *KDD*, 2009.

[23] Z. Zhao, L. Wang, and H. Liu, “Efficient spectral feature selection with minimum redundancy,” in *AAAI*, 2010.

[24] Z. Zhao and H. Liu, “Spectral feature selection for supervised and unsupervised learning,” in *ICML*, 2007.

[25] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, “Liblinear: A library for large linear classification,” in *JMLR*, 2008.

[26] R. Duda, P. Hart, D. Stork *et al.*, in *Pattern classification.*wiley New York, 2001.

[27] Q. Gu, Z. Li, and J. Han, “Generalized fisher score for feature selection,” in *UAI*, 2011.

[28] J. Olvera-López, J. Carrasco-Ochoa, J. Martínez-Trinidad, and J. Kittler, “A review of instance selection methods,” in *Artificial Intelligence Review*, 2010.

[29] H. Liu and L. Yu, “Toward integrating feature selection algorithms for classification and clustering,” in *TKDE*, 2005.

[30] M. Robnik-Šikonja and I. Kononenko, “Theoretical and empirical analysis of ReliefF and RReliefF,” in *Machine learning*, 2003.

[31] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy,” in *TPAMI*, 2005.

[32] X. He, D. Cai, and P. Niyogi, “Laplacian score for feature selection,” in *NIPS*, 2006.

[33] X. Hu, L. Tang, J. Tang and H. Liu, “Exploiting Social Relations for Sentiment Analysis in Microblogging,” in *WSDM*, 2013.

[34] J. Dy and C. Brodley, “Feature selection for unsupervised learning,” in *JMLR*, 2004.

[35] C. Constantinopoulos, M. Titsias, and A. Likas, “Bayesian feature and model selection for gaussian mixture models,” in *TPAMI*, 2006.

[36] G. Cawley, N. Talbot, and M. Girolami, “Sparse multinomial logistic regression via bayesian l1 regularisation,” in *NIPS*, 2006.

[37] A. Argyriou, T. Evgeniou, and M. Pontil, “Multi-task feature learning,” in *NIPS*, 2007.

[38] H. Gao, G. Barbier, and R. Gollsby, “Harnessing the crowdsourcing power of social media for disaster relief,” in *IEEE Intelligent Systems*, 2011.

[39] X. Hu, N. Sun, C. Zhang and T. Chua, “Exploiting internal and external semantics for the clustering of short texts using world knowledge,” in *CIKM*, 2009.

[40] J. He and J. Carbonell, “Coselection of features and instances for unsupervised rare category analysis,” in *Statistical Analysis and Data Mining*, 2010.

[41] R. Xiang, J. Neville, and M. Rogati, “Modeling relationship strength in online social networks,” in *WWW*, 2010.