Unsupervised Feature Selection for Multi-View Data in Social Media

Jiliang Tang* Xia Hu* Huiji Gao* Huan Liu*

Abstract

The explosive popularity of social media produces mountains of high-dimensional data and the nature of social media also determines that its data is often unlabelled, noisy and partial, presenting new challenges to feature selection. Social media data can be represented by heterogeneous feature spaces in the form of multiple views. In general, multiple views can be complementary and, when used together, can help handle noisy and partial data for any single-view feature selection. These unique challenges and properties motivate us to develop a novel feature selection framework to handle multi-view social media data. In this paper, we investigate how to exploit relations among views to help each other select relevant features, and propose a novel unsupervised feature selection framework, MVFS, for multi-view social media data. We systematically evaluate the proposed framework in multi-view datasets from social media websites and the results demonstrate the effectiveness and potential of MVFS.

1 Introduction

The prevalent use of social media generates massive data at an unprecedented rate. For example, 250 million tweets are posted per day[1]; 3,000 photos are uploaded per minute to Flickr[2]; and 60 hours of video are uploaded every minute to Youtube[3]. Such data can be usually represented by heterogeneous feature spaces in the form of multiple views. For example, when users upload photos into Flickr, they are asked to provide tags as well as text descriptions about the photos. Thus photos can be described by three high-dimensional feature spaces, i.e., SIFT feature space (visual photo content) [1], tag space (tags), and term space (text descriptions). The nature of social media also determines that each view is often noisy and incomplete. The multi-view data raises a natural, yet rarely exploited problem: how to prepare the high-dimensional multi-view data for effective data mining.

Feature selection has been proven to be an effective approach to handling large-scale and high-dimensional data [22, 5]. Most of the existing feature selection methods were developed for traditional single-view data. Two straightforward strategies to apply existing feature selection methods to multi-view social media data are: (1) the concatenating strategy - converting multi-view data into single-view data by concatenating heterogeneous feature spaces into one homogeneous feature space as shown in Figure 1(a); and (2) the separation strategy - performing traditional feature selection on each view separately as demonstrated in Figure 1(b). The concatenating strategy ignores the differences among heterogeneous feature spaces while the separation strategy considers each view independently. However, views are inherently related since they describe the same set of objects through different feature spaces. In general, multiple views can provide complementary information such as tags and text descriptions in Flickr provide semantic information about photos, and can potentially help us achieve better performance. Note that we do not consider the concatenating strategy in our current work since our initial experimental results show that its performance is much worse than that of the separation strategy and we call traditional feature selection with the separation strategy as *single-view feature selection* for convenience.

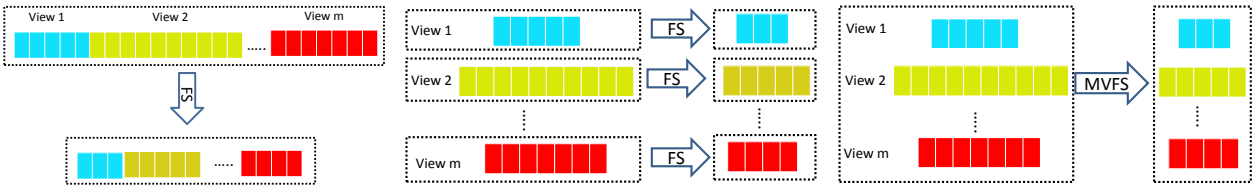
In this paper, we study a novel problem of feature selection for multi-view social media data in an unsupervised scenario as in Figure 1(c): views are represented by heterogeneous feature spaces and it aims to select features for all views simultaneously by exploiting relations among views. For example, we select visual features, tags, and terms for photos in Flickr simultaneously. This multi-view feature selection problem is apparently distinct from single-view feature selection: (1) multi-view feature selection studies multiple views together and exploits relations among views while single-view feature selection studies each view separately; and (2) multi-view feature selection can select features from heterogeneous feature spaces simultaneously while single-view feature selection selects features from a homogeneous feature space each time. Since single-view feature selection algorithms are unequipped for multi-view social media data, we propose to inves-

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

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

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

[3]http://www.youtube.com/t/press_statistics



(a) The Concatenating Strategy (b) The Separation Strategy (c) Multi-view Feature Selection

Figure 1: Different Strategies to Handle Multi-view Social Media Data

tigate two problems: (1) how to exploit the relations among views; and (2) how to take advantage of these relations for unsupervised feature selection. The solutions to the two problems result in an unsupervised feature selection framework MVFS for multi-view data. The main contributions of this paper are summarized below,

- Defining the novel problem of unsupervised feature selection for multi-view data formally and proposing to exploit the relations among views in formulating multi-view feature selection;
- Introducing a new way to capture the relations among views and guide the development of an unsupervised feature selection framework;
- Proposing a novel unsupervised feature selection framework, MVFS, for multi-view data to select features from heterogeneous feature spaces simultaneously by exploiting view relations; and
- Evaluating the proposed framework, MVFS, systematically using datasets from real-world social media websites in comparison with single-view feature selection algorithms.

2 Problem Statement

Let $\mathbf{p} = \{p_1, p_2, \ldots, p_n\}$ be the set of posts where n is the number of posts. Note that we use post here in a loose way to cover user generated content in social media such as photos, videos, or tweets. Assume that the set of posts i.e., $\mathbf{p}$, can be represented by m heterogeneous feature spaces with m views. Let $\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_m\}$ be the set of m feature spaces and $\mathbf{f}_i \in \mathbb{R}^{\ell_i}$ denote the feature space for the i-th view where ℓ_i is the number of features in the i-th feature space $\mathbf{f}_i$. Let $\mathcal{X} = \{\mathbf{X}_i \in \mathbb{R}^{\ell_i \times n}\}_{i=1}^m$ denote the set of views and $\mathbf{X}_i$ be the matrix representation of the i-th view.

With the notations defined above, the problem of *unsupervised feature selection for multi-view data* can be stated as: given n objects with multi-view representations $\mathcal{X}$ by m heterogeneous feature spaces $\mathcal{F}$, develop a method H_{MVFS} which can select a subset of relevant features (e.g., $\mathbf{f}_i'$), for each feature space (e.g., $\mathbf{f}_i$), simultaneously for these n objects by exploiting the relations among views.

3 Unsupervised Feature Selection Framework for Multi-View Data

Recall that multi-view data in social media poses two main challenges for unsupervised feature selection: (1) how to exploit relations among views; and (2) how to take advantage of these relations for unsupervised feature selection. In the following subsections, we will present the details about the solutions to these two challenges based on pseudo-class labels, resulting in a novel unsupervised feature selection framework for multi-view data.

3.1 Exploiting Relations among views

Most existing work about multi-view learning assumes that all views share the same label space, and the views are correlated through the label space [6]. As we know, the main difficulty with unsupervised feature selection is due to the lack of class labels. To tackle the challenges caused by the lack of class labels, we introduce the concept of pseudo-class labels to exploit relations among views, guiding the development of the framework.

We assume that $\mathbf{Z} \in \mathbb{R}^{n \times k}$ is the pseudo-class label matrix where k is the number of pseudo-class labels and each data point belongs to only one class where $\mathbf{Z}(i,j) = 1$ if p_i belongs to the j-th pseudo-class, otherwise $\mathbf{Z}(i,j) = 0$. Thus $\mathbf{Z}$ should satisfy the following constraints,

$$\mathbf{Z}(i,:) \in \{0,1\}^n, \quad \|\mathbf{Z}(i,:)\|_0 = 1, \quad \forall\, i,\ 1 \le i \le n,$$

where $\|\cdot\|_0$ is the vector zero norm, which counts the number of nonzero elements in the vector.

Set $\mathbf{s}_i = \pi(\overbrace{0, \ldots, 0}^{\ell_i - k_i}, \overbrace{1, \ldots, 1}^{k_i})$ where $\pi(\cdot)$ is the permutation function and k_i is the number of features to select for the i-th view where $\mathbf{s}_i(j) = 1$ indicates that the j-th feature in the i-th feature space $\mathbf{f}_i$ is selected. The original view $\mathbf{X}_i$ can be represented as $\mathbf{X}_i' = \text{diag}(\mathbf{s}_i)\mathbf{X}_i$ with k_i selected features, where $\text{diag}(\mathbf{s}_i)$ is a diagonal matrix. With the pseudo-class label information, we further assume that there is a

mapping matrix $\mathbf{W}_i \in \mathbb{R}^{\ell_i \times k}$ for the i-th view, which assigns data points with pseudo-class labels. Let $\mathbf{Y}_i = (\mathbf{X}_i')^\top \mathbf{W}_i$ be the pseudo-class label assignment matrix and $\mathcal{W} = \{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_m\}$ be the set of mapping matrices. Then the relations among views can be captured by the following optimization problem,

$$\begin{aligned} \min_{\mathcal{W},\mathbf{Z}} \quad & \sum_{i=1}^{m} \|(\mathbf{X}_i')^\top \mathbf{W}_i - \mathbf{Z}\|_F^2, \\ s.t. \quad & \|\mathbf{Z}(i,:)\|_0 = 1, \quad i \in \{1,2,\ldots,n\}, \\ & \mathbf{Z}(i,j) \in \{0,1\}, \quad j \in \{1,2,\ldots,k\}, \end{aligned} \tag{3.1}$$

where we force each pseudo-class label assignment matrix $\mathbf{Y}_i$, close to the pseudo-class label matrix $\mathbf{Z}$, according to the assumption that views are correlated through their shared label space.

3.2 The Framework: MVFS With pseudo-class labels, we are further allowed to take advantage of information from each view, i.e., $\mathbf{X}_i$, based on spectral analysis [26]: similar data instances should have similar labels. Thus the constraint from the i-th view can be formulated as the following minimization problem,

$$\min \quad Tr(\mathbf{Z}^\top \mathbf{L}_i \mathbf{Z}) \tag{3.2}$$

where $\mathbf{L}_i = \mathbf{V}_i - \mathbf{S}_i$ is a Laplacian matrix and $\mathbf{V}_i$ is a diagonal matrix with its elements defined as $\mathbf{V}_i(j,j) = \sum_{K=1}^{n} \mathbf{S}_i(K,j)$. $\mathbf{S}_i \in \mathbb{R}^{n \times n}$ denotes the similarity matrix based on $\mathbf{X}_i$ via a RBF kernel in this work.

By introducing the concept of pseudo-class labels, we can exploit the relations among views as well as the information from each view. With these preliminary solutions, we propose a novel unsupervised feature selection framework, MVFS, for multi-view data, which is formulated to solve the following optimization problem.

$$\begin{aligned} \min_{\mathcal{W},\mathbf{Z},\mathbf{s}_i} \quad & \sum_{i=1}^{m} \lambda_i \big(Tr(\mathbf{Z}^\top \mathbf{L}_i \mathbf{Z}) + \alpha \|(\mathbf{X}_i')^\top \mathbf{W}_i - \mathbf{Z}\|_F^2 \big) \\ s.t. \quad & \mathbf{s}_i \in \{0,1\}^n, \ \mathbf{s}_i^\top \mathbf{1}_n = k_i, \\ & \|\mathbf{Z}(i,:)\|_0 = 1, \quad i \in \{1,2,\ldots,n\}, \\ & \mathbf{Z}(i,j) \in \{0,1\}, \quad j \in \{1,2,\ldots,k\}. \end{aligned} \tag{3.3}$$

In Eq. (3.3), the first term is used to obtain the information from each view while the second term is used to exploit the relations among views by their shared label space, i.e., pseudo-class label $\mathbf{Z}$. α is introduced to control the contributions of these two parts. The parameter λ_i is employed to control the contributions from each view and $\sum_{i=1}^{m} \lambda_i = 1$.

The constraints in Eq. (3.3), mixed vector zero norm with integer programming, make the problem difficult to solve. First, we consider the constraints on the pseudo-class label indicator matrix $\mathbf{Z}$. By relaxing the value of $\mathbf{Z}$ from $\{0,1\}$ to a continuous nonnegative value, we convert the constraints on $\mathbf{Z}$ into the constraints,

$$\mathbf{Z}^\top \mathbf{Z} = \mathbf{I}_k, \quad \mathbf{Z} \geq 0 \tag{3.4}$$

where the orthogonal and nonnegative constraints on $\mathbf{Z}$ guarantee that there is only one positive element in each row and others are zeros.

Because vectors of zero norm are mixed with integer programming in the constraints of Eq. (3.3), they make the problem difficult to solve. We observe that $\mathrm{diag}(\mathbf{s}_i)$ and $\mathbf{W}_i$ is always in the form of $\mathrm{diag}(\mathbf{s}_i)\mathbf{W}_i$ in Eq. (3.3). Since $\mathbf{s}_i$ is a binary vector and $\ell_i - k_i$ rows of $\mathrm{diag}(\mathbf{s}_i)$ are all zeros, $\mathrm{diag}(\mathbf{s}_i)\mathbf{W}_i$ is a matrix where the elements of many rows are all zeros. This motivates us to absorb $\mathrm{diag}(\mathbf{s}_i)$ into $\mathbf{W}_i$, $\mathbf{W}_i = \mathrm{diag}(\mathbf{s}_i)\mathbf{W}_i$, and add $\ell_{2,1}$ norm on $\mathbf{W}_i$ to ensure the sparsity of $\mathbf{W}_i$ in rows and achieve feature selection.

With these relaxations, MVFS is to solve the following optimization problem,

$$\begin{aligned} \min_{\mathcal{W},\mathbf{Z}} \quad & \mathcal{J}(\mathcal{W},\mathbf{Z}) = \sum_{i=1}^{m} \lambda_i \big(Tr(\mathbf{Z}^\top \mathbf{L}_i \mathbf{Z}) + \\ & \alpha(\|\mathbf{X}_i^\top \mathbf{W}_i - \mathbf{Z}\|_F^2 + \beta \|\mathbf{W}_i\|_{2,1}) \big) \\ s.t. \quad & \mathbf{Z}^\top \mathbf{Z} = \mathbf{I}, \qquad \mathbf{Z} \geq 0. \end{aligned} \tag{3.5}$$

where $\|\mathbf{W}_i\|_{2,1}$ controls the capacity of $\mathbf{W}_i$ and also ensures that $\mathbf{W}_i$ is sparse in rows, making it particularly suitable for feature selection. The parameter β controls the sparsity of $\mathbf{W}_i$.

There are $m+1$ components, i.e., $\mathcal{W}$ and $\mathbf{Z}$, in the objective function of Eq. (3.5), and it is difficult to optimize these components simultaneously. Below, we apply an alternating optimization to solve this problem and update $\{\mathbf{W}_i\}_{i=1}^m$ and $\mathbf{Z}$ iteratively and alternatingly to find an optimal solution for Eq. (3.5).

Computing $\{\mathbf{W}_i\}_{i=1}^m$, given $\mathbf{Z}$: If $\mathbf{Z}$ is fixed, the constraints are independent on $\{\mathbf{W}_i\}_{i=1}^m$ and the relations among views are decoupled, suggesting that we can optimize each $\mathbf{W}_i$ $(1 \leq i \leq m)$, separately. $\mathbf{W}_i$ can be obtained by solving the following problem,

$$\min_{\mathbf{W}_i} \ \mathcal{J}(\mathbf{W}_i) = \|\mathbf{X}_i^\top \mathbf{W}_i - \mathbf{Z}\|_F^2 + \beta \|\mathbf{W}_i\|_{2,1}. \tag{3.6}$$

Taking the derivation of $\mathcal{J}(\mathbf{W}_i)$ and setting it to zero, we can obtain the update rule for $\mathbf{W}_i$ as,

$$\mathbf{W}_i = \big(\mathbf{X}_i \mathbf{X}_i^\top + \beta \mathbf{D}_i\big)^{-1} \mathbf{X}_i \mathbf{Z}, \tag{3.7}$$

where $\mathbf{D}_i$ is a diagonal matrix with the j-th diagonal element is $\mathbf{D}_i(j,j) = \frac{1}{2\|\mathbf{W}_i(j,:)\|_2}$. We develop the

following theorem to show that the update rule in Eq. (3.7) can monotonically decrease the objective function value of $\mathcal{J}(\mathbf{W}_i)$.

THEOREM 3.1. *The update rule in Eq. (3.7) can monotonically decrease the objective value of* $\mathcal{J}(\mathbf{W}_i)$.

Proof. The proof process is similar to that in [27, 29]. To save space, we ignore the detailed proof.

Computing $\mathbf{Z}$, given $\{\mathbf{W}\}_{i=1}^m$: If $\{\mathbf{W}\}_{i=1}^m$ are fixed, $\mathbf{Z}$ can be obtained through solving the following optimization problem,

$$\min_{\mathbf{Z}} \quad \mathcal{J}(\mathbf{Z}) = Tr(\mathbf{Z}^\top \mathbf{M}\mathbf{Z}) + \sum_{i=1}^m \gamma_i \|\mathbf{A}_i - \mathbf{Z}\|_F^2$$
$$s.t. \quad \mathbf{Z}^\top \mathbf{Z} = \mathbf{I}, \qquad \mathbf{Z} \geq 0. \tag{3.8}$$

where $\mathbf{M} = \sum_{i=1}^m \lambda_i \mathbf{L}_i$, $\gamma_i = \lambda_i \alpha$ and $\mathbf{A}_i = \mathbf{X}_i^\top \mathbf{W}_i$.

The Lagrangian function of Eq. (3.8) is:

$$\mathcal{L}(\mathbf{Z}) = Tr(\mathbf{Z}^\top \mathbf{M}\mathbf{Z}) + Tr\big(\Gamma(\mathbf{Z}^\top \mathbf{Z} - \mathbf{I})\big)$$
$$- Tr(\Lambda \mathbf{Z}) + \sum_{i=1}^m \gamma_i Tr(-2\mathbf{A}_i^\top \mathbf{Z} + \mathbf{Z}^\top \mathbf{Z}), \tag{3.9}$$

where Γ and Λ are Lagrangian multipliers. Using the KKT condition, $\Lambda(j,l)\mathbf{Z}(j,l) = 0$, we obtain,

$$(\mathbf{M}\mathbf{Z} + \sum_{i=1}^m \big(-\gamma_i \mathbf{A}_i + \gamma_i \mathbf{Z}\big) + \mathbf{Z}\Gamma)(j,l)\mathbf{Z}(j,l) = 0.$$

which leads to the following update rule for $\mathbf{Z}$,

$$\mathbf{Z}(j,l) \leftarrow \mathbf{Z}(j,l)\sqrt{\frac{\big(\mathbf{M}^-\mathbf{Z} + \sum_{i=1}^m \gamma_i \mathbf{A}_i^+ + \mathbf{Z}\Gamma^-\big)(j,l)}{\big(\mathbf{M}^+\mathbf{Z} + \sum_{i=1}^m \gamma_i(\mathbf{A}_i^- + \mathbf{Z}) + \mathbf{Z}\Gamma^+\big)(j,l)}} \tag{3.10}$$

where $\mathbf{X}^+(j,l) = (|\mathbf{X}(j,l)| + \mathbf{X}(j,l))/2$, $\mathbf{X}^-(j,l) = (\mathbf{X}(j,l) - |\mathbf{X}(j,l)|)/2$ and $\mathbf{X} = \mathbf{X}^+ - \mathbf{X}^-$.

From Eq. (3.9), summing over j, we obtain $\Gamma(j,j) = \big(\sum_{i=1}^m \gamma_i(\mathbf{Z}^\top \mathbf{A}_i - \mathbf{I}) - \mathbf{Z}^\top \mathbf{M}\mathbf{Z}\big)(j,j)$. The off-diagonal elements of Γ are approximately obtained by ignoring the non-negative of $\mathbf{Z}$, which leads to $\Gamma(j,l) = \big(\sum_{i=1}^m \gamma_i(\mathbf{Z}^\top \mathbf{A}_i - \mathbf{I}) - \mathbf{Z}^\top \mathbf{M}\mathbf{Z}\big)(j,l)$. Combining these, we get $\Gamma = \sum_{i=1}^m \gamma_i(\mathbf{Z}^\top \mathbf{A}_i - \mathbf{I}) - \mathbf{Z}^\top \mathbf{M}\mathbf{Z}$.

Next we will use the auxiliary function approach [10] to show that the update rule in Eq. (3.10) will monotonically decrease the value of the objective in Eq. (3.8). The definition of the auxiliary function can be found in [10].

THEOREM 3.2. *Let*

$$H(\mathbf{Z}) = Tr\big(\mathbf{Z}^\top \mathbf{M}\mathbf{Z} + \sum_{i=1}^m \gamma_i(-2\mathbf{A}_i^\top \mathbf{Z} + \mathbf{Z}^\top \mathbf{Z}) + \Gamma(\mathbf{Z}^\top \mathbf{Z} - \mathbf{I})\big).$$

Then the following function $h(\mathbf{Z}, \tilde{\mathbf{Z}})$,

$$h(\mathbf{Z}, \tilde{\mathbf{Z}}) = \sum_{ijl} \gamma_i \big(\mathbf{A}_i^-(j,l)\frac{\mathbf{Z}^2(j,l) + \tilde{\mathbf{Z}}^2(j,l)}{\tilde{\mathbf{Z}}(j,l)} + \frac{\tilde{\mathbf{Z}}(j,l)\mathbf{Z}^2(j,l)}{\tilde{\mathbf{Z}}(j,l)}\big)$$
$$+ \sum_{jl} \big(\frac{(\tilde{\mathbf{Z}}\Gamma^+)(j,l)\mathbf{Z}^2(j,l)}{\tilde{\mathbf{Z}}(j,l)} + \frac{(\mathbf{M}^+\tilde{\mathbf{Z}})(j,l)\mathbf{Z}^2(j,l)}{\tilde{\mathbf{Z}}(j,l)}\big)$$
$$- \Big(\sum_{jl}\big(\sum_i 2\gamma_i \mathbf{A}_i(j,l)\big)\tilde{\mathbf{Z}}(j,l)\big(1 + \log\frac{\mathbf{Z}(j,l)}{\tilde{\mathbf{Z}}(j,l)}\big)$$
$$+ \sum_{ijl} \Gamma^-(j,l)\tilde{\mathbf{Z}}(i,j)\tilde{\mathbf{Z}}(i,k)\big(1 + \log\frac{\mathbf{Z}(i,j)\mathbf{Z}(i,l)}{\tilde{\mathbf{Z}}(i,j)\tilde{\mathbf{Z}}(i,k)}\big)$$
$$+ \sum_{ijl} \mathbf{M}^-(j,l)\tilde{\mathbf{Z}}(j,i)\tilde{\mathbf{Z}}(k,i)\big(1 + \log\frac{\mathbf{Z}(j,k)\mathbf{Z}(i,l)}{\tilde{\mathbf{Z}}(j,i)\tilde{\mathbf{Z}}(j,l)}\big)\Big),$$

is an auxiliary function of $H(\mathbf{Z})$. $h(\mathbf{Z}, \tilde{\mathbf{Z}})$ *is convex and its global minimum is,*

$$\mathbf{Z}(j,l) = \mathbf{Z}(j,l)\sqrt{\frac{\big(\mathbf{M}^-\mathbf{Z} + \sum_{i=1}^m \gamma_i \mathbf{A}_i^+ + \mathbf{Z}\Gamma^-\big)(j,l)}{\big(\mathbf{M}^+\mathbf{Z} + \sum_{i=1}^m \gamma_i(\mathbf{A}_i^- + \mathbf{Z}) + \mathbf{Z}\Gamma^+\big)(j,l)}}.$$

Proof. The detailed proof is presented in Appendix 6.1.

THEOREM 3.3. *Updating* $\mathbf{Z}$ *as Eq. (3.10) will monotonically decrease the value of* $\mathcal{J}(\mathbf{Z})$.

Proof. Since $H(\mathbf{Z})$ is the Lagrangian function of Eq. (3.8) with KKT condition, we just need to verify that the update rule will monotonically decrease the value of $H(\mathbf{Z})$. Through the definition of the auxiliary function and Theorem 3.2, we can obtain the following inequality chain:

$$H(\mathbf{Z}^0) = h(\mathbf{Z}^0, \mathbf{Z}^0) \geq h(\mathbf{Z}^0, \mathbf{Z}^1) \geq H(\mathbf{Z}^1) \ldots$$

, which completes the proof.

With update rules for both $\mathcal{W}$ and $\mathbf{Z}$, we present the detailed algorithm to optimize the problem defined in Eq. (3.5) in Algorithm 1. For the convergence of the proposed algorithm, we develop the following theorem.

THEOREM 3.4. *Algorithm 1, optimizing the objective function* $\mathcal{J}(\mathcal{W}, \mathbf{Z})$, *converges.*

Proof. With Theorem 3.1 and Theorem 3.3, we can get the following inequality chain,

$$\mathcal{J}(\mathcal{W}_0, \mathbf{Z}_0) \geq \mathcal{J}(\mathcal{W}_1, \mathbf{Z}_0) \geq \mathcal{J}(\mathcal{W}_1, \mathbf{Z}_1) \ldots.$$

$\mathcal{J}(\mathcal{W}, \mathbf{Z})$ is bounded since $\mathcal{J}(\mathcal{W}, \mathbf{Z}) \geq 0$. Thus Algorithm 1 converges, which completes the proof.

Algorithm 1 The Proposed Framework: MVFS

Input: $\{\mathbf{X}_i, \lambda_i, k_i\}_{i=1}^m$, k, α, β
Output: $k_i (1 \leq i \leq m)$ features for the i-th view

1: **for** $i = 1$ to m **do**
2: Construct the laplacian matrix $\mathbf{L}_i$
3: Set $\mathbf{D}_i$ as an identify matrix and set $\gamma_i = \lambda_i \alpha$
4: **end for**
5: Construct $\mathbf{M} = \sum_{i=1}^m \lambda_i \mathbf{L}_i$, $\mathbf{M}^+$ and $\mathbf{M}^-$
6: Initialize $\mathbf{Z}$ as $\mathbf{Z}^\top \mathbf{Z} = \mathbf{I}$
7: **while** Not convergent **do**
8: **for** $i = 1$ to m **do**
9: Update $\mathbf{W}_i \leftarrow \left(\mathbf{X}_i \mathbf{X}_i^\top + \beta \mathbf{D}_i\right)^{-1} \mathbf{X}_i \mathbf{Z}$
10: Construct $\mathbf{A}_i = \mathbf{X}_i^\top \mathbf{W}_i$, $\mathbf{A}_i^+$ and $\mathbf{A}_i^-$
11: **end for**
12: Set $\Gamma = \sum_{i=1}^m \gamma_i (\mathbf{Z}^\top \mathbf{A}_i - \mathbf{I}) - \mathbf{Z}^\top \mathbf{M} \mathbf{Z}$, Γ^+ and Γ^-
13: Update $\mathbf{Z}$ via Eq. (3.10)
14: **end while**
15: **for** $i = 1$ to m **do**
16: Sort each feature for $\mathbf{X}_i$ according to $\|\mathbf{W}_i(j,:)\|_2$ in descending order and select the top-k_i ranked ones.
17: **end for**

4 Experiments

In this section, we conduct experiments to evaluate our proposed framework by answering the following two questions: (1) do we gain by studying MVFS as expected? and (2) which strategy is better for multi-view data? Since we only consider the separation strategy for applying existing feature selection methods to multi-view data, we can answer both questions by studying the effectiveness of the proposed framework comparing with the state-of-the-art single-view feature selection algorithms. Finally we investigate how the parameters of MVFS (e.g., numbers of pseudo-class labels) affect feature selection performance.

4.1 Datasets To evaluate MVFS, we crawled two multi-view datasets from real-world social media websites, i.e., Flickr and BlogCatalog.

Flickr is a photo sharing website where users can specify tags and provide text descriptions for photos they upload. Photos are organized under pre-specified categories, used as the ground truth of class labels in our experiment. Each post in this dataset has three views, i.e., photo visual content (V_1), tags (V_2) and text descriptions (V_3).

BlogCatalog[4] is a blog directory website where users can register their blogs under predefined categories, used as the ground truth of class labels in our evaluation. To improve the access to blogs, users also specify the tags associated with each blog. Therefore each post has two views, i.e., blog content (V_1) and its associated tags (V_2).

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

Table 1: Statistics of the Datasets

	Flickr			BlogCatalog	
# Posts	379			3,744	
# Classes	6			6	
# Views	3			2	
Views	V_1	V_2	V_3	V_1	V_2
# Features	7,500	3,631	4,570	6,115	5,764

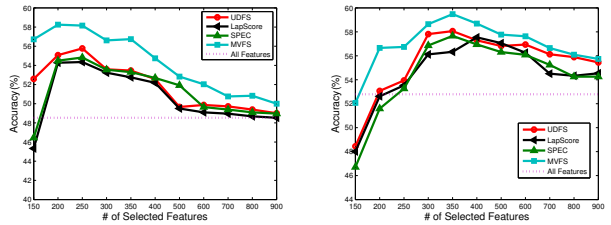
Posts, represented by term space, are preprocessed for stop-word removal and stemming. Some statistics of these datasets are shown in Table 1.

4.2 Baseline Methods and Evaluation Metrics MVFS is compared with the following representative unsupervised single-view feature selection algorithms: (1) LapScore [20] - the importance of a feature is evaluated via its power of locality preservation; (2) SPEC [33] - features are selected through spectral regression; and (3) UDFS [32] - features are selected in batch mode by simultaneously exploiting feature correlation and discriminative information. The focus of this paper is to investigate whether we can gain from the study of multi-view feature selection. After the verification of its effectiveness, we can further explore the effect of link information [29, 30] on multi-view feature selection.

Following the usual evaluation way for unsupervised feature selection, we assess the proposed framework in terms of clustering performance (both single-view clustering and multi-view clustering). Two common used metrics are adopted to evaluate the clustering quality, i.e., *normalized mutual information* (NMI) and *accuracy*. We vary the numbers of selected features as $\{150, 200, 250, 300, 350, 400, 500, 600, 700, 800, 900\}$.

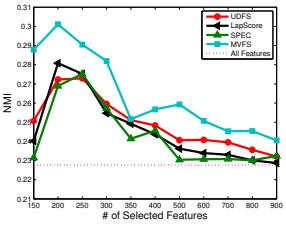
4.3 Feature Selection for Single-view Clustering In this subsection, we investigate how different feature selection methods affect the performance of single-view clustering. Each feature selection algorithm is first performed to select features, and then a representative single-view clustering method, K-means, 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.

For each dataset, MVFS can select features for all views simultaneously while baseline methods are single-view algorithms and select features for each view separately. Conventionally, the parameters in feature selection algorithms are tuned via cross-validation.

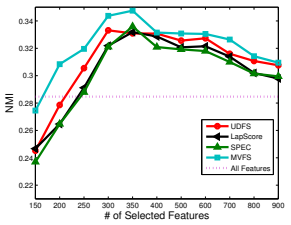


(a) Accuracy in V_1

(b) Accuracy in V_2



(c) NMI in V_1



(d) NMI in V_2

Figure 3: Single-view Clustering Performance with Different Feature Selection Algorithms in BlogCatalog. Note that dash lines denote the clustering performance with all features for each view.

For MVFS, λ_is are coefficients to combine multiple views, which can be learnt automatically from data [7] and other parameters are determined through cross-validation. More details about the parameter selection for MVFS are given in Section 4.5. The resulting parameter values for MVFS are: $\{(\lambda_1 = 0.4, \lambda_2 = 0.3, \lambda_3 = 0.3), \alpha = 0.1, \beta = 0.1, k = 6\}$ for Flickr while $\{(\lambda_1 = 0.6, \lambda_2 = 0.4), \alpha = 0.3, \beta = 0.1, k = 6\}$ for BlogCatalog. The comparison results in Flickr and BlogCatalog are shown in Figures 2 and 3, respectively. Note that dash lines in figures denote the clustering performance with all features for each view.

Most of the time, with an increasing number of features, the performance in terms of both Accuracy and NMI first increases rapidly, reaches its peak value and then degrades gradually. With smaller numbers of selected features, we lose so much information about the original data while we introduce irrelevant features with larger numbers of selected features.

LapScore obtains comparable results with SPEC in both datasets. Most of time, UDFS achieves slightly better performance than LapScore and SPEC. LapScore and SPEC analyze features separately and select features one after another while UDFS selects features in a batch mode and considers feature correlation [32].

We observe that our proposed MVFS algorithm consistently outperforms all baseline methods. For example, MVFS obtains up to 16.81% and 14.80% relative improvement w.r.t. NMI in Flickr and BlogCatalog, respectively. Multiple views provide complementary information and can help each other to select relevant features. We also note that even for the same dataset, the improvement of MVFS is different for different views. For example, in BlogCatalog, on average, MVFS obtains 4.39% relative improvement with respect to NMI in V_1 while it gains 2.69% improvement in V_2.

Table 2: The Performance of Multi-view Clustering

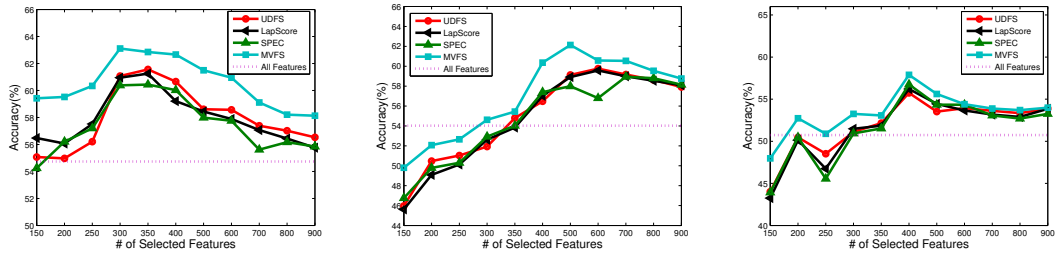
Datasets		Flickr		BlogCatalog	
		Accuracy	NMI	Accuracy	NMI
All Features		56.92	0.3814	55.06	0.3288
F-Optimal	LapScore	63.31	0.4601	60.42	0.3787
	SPEC	63.54	0.4579	60.66	0.3769
	UDFS	64.02	0.4601	59.98	0.3797
	MVFS	66.85	0.4877	63.04	0.3963
F-Min	LapScore	57.99	0.4021	55.84	0.3204
	SPEC	57.73	0.3959	55.84	0.3229
	UDFS	58.09	0.3959	57.45	0.3229
	MVFS	63.64	0.4355	61.57	0.3594
F-Max	LapScore	59.03	0.4009	58.03	0.3457
	SPEC	59.42	0.4021	58.31	0.3475
	UDFS	59.99	0.4077	58.47	0.3492
	MVFS	64.08	0.4388	61.74	0.3686

4.4 Feature Selection for Multi-view Clustering

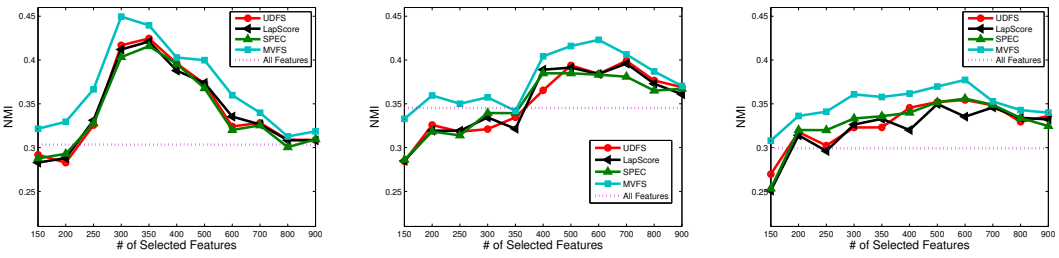
In this subsection, we investigate how feature selection methods affect multi-view clustering. The experimental settings are almost the same as single-view clustering except the clustering method and the number of selected features. For this evaluation, multi-view clustering replaces single-view clustering as the clustering method. Corresponding to K-means, the multi-view version of K-means [3] is chosen for multi-view clustering.

Determining the optimal number of selected features for single-view clustering is an open problem [32] and it is even more difficult for multi-view clustering considering the combinations of multiple views. For a fair comparison, we use three ways to choose the number of selected features: (1) F-Optimal: the number for a specific feature selection method in each view is chosen as when the best performance is achieved in the single-view clustering experiments. For example, according to Figure 3, the numbers of selected features for UDFS are chosen to be 250 and 300 for V_1 and V_2 in BlogCatalog, respectively; (2) F-Min: the number is fixed to 150, which is the minimal number of selected features in single-view clustering experiments; (3) F-Max: the number is fixed to 900, which is the maximal number of selected features in single-view clustering experiments. The results are demonstrated in Table 2.

The first observation is that the multi-view version of K-means (mvK-Means) significantly improves on its single-view counterpart (K-means). For example, when using all features, mvK-Means obtains 25.71%, 10.45%, 27.52% improvement in terms of NMI for V_1, V_2 and V_3 of Flickr, respectively. We note that for these



(a) Accuracy in V_1 (b) Accuracy in V_2 (c) Accuracy in V_3



(d) NMI in V_1 (e) NMI in V_2 (f) NMI in V_3

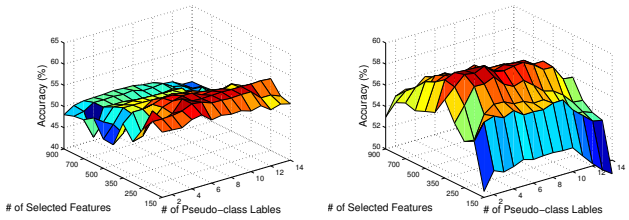
Figure 2: Single-view Clustering Performance with Different Feature Selection Algorithms in Flickr. Note that dash lines denote the clustering performance with all features for each view.

three ways, MVFS always obtains the best performance. With the single-view clustering experiments, we can conclude that *we obtain performance improvement in terms of both single-view and multi-view clustering by the study of multi-view feature selection as expected.*

4.5 Parameter Selection There are two important parameters of the proposed framework, MVFS, i.e., k the number of pseudo-class labels and α controlling the contribution of capturing the relations among views. How to determine the optimal number of selected features is still an open question. In the following subsection, we will systematically investigate how the performance of MVFS varies with its parameters (k or α) and the number of selected features in terms of single-view clustering. To save space, we only show the results in BlogCatalog w.r.t. Accuracy since we have similar observations with other settings.

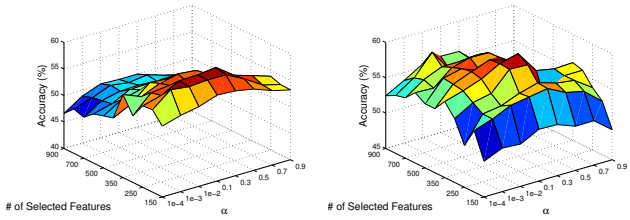
Setting $\alpha = 0.3$, we vary k from 2 to 14 with an incremental step of 1 and the performance variation w.r.t. k and the number of selected features is shown in Figure 4. We note that with the increase of k, the performance first increases, achieves its peak value and then decreases. This observation can be used to determine the optimal number of k. We also observe that the performance is not sensitive to k when k is from 4 to 10.

Setting $k = 6$, we vary α as $\{1e{-}4, 1e{-}3, 1e{-}2, 0.1, 0.3, 0.5, 0.7, 0.9\}$. The performance variance w.r.t. α and the number of features is depicted in Figure 5. Most



(a) V_1 (b) V_2

Figure 4: Numbers of Pseudo-class Labels vs Numbers of Selected Features



(a) V_1 (b) V_2

Figure 5: α vs Numbers of Selected Features

of the time, MVFS achieves its best performance when $\alpha = 0.3$, indicating the importance of capturing the relations among views. After that, the performance decreases, suggesting the importance of information from each view. It also shows that an appropriate combination of these two components is crucial for MVFS to improve the performance.

5 Related Work

Based on whether the training data is labeled or not, feature selection algorithms can be either supervised or unsupervised [13, 23, 5]. Supervised feature selection methods assess feature relevance by label information. It can be further categorized into the *wrapper* models [14, 21] and the *filter* models [23, 28]. Due to the availability of a large amount of unlabeled data, unsupervised feature selection attracts more and more attention [31, 20, 9, 5]. One key difference between supervised and unsupervised feature selection is the availability of label information. Without label information that guides to access feature relevance, unsupervised feature selection [20, 13, 33] is particularly difficult since it is a less constrained search problem, depending on clustering quality measures [16, 15], and can eventuate many equally valid feature subsets. Without considering additional constraints, it is likely to find many equally good sets of features for high-dimensional data. Another key difficulty is how to objectively measure the results of feature selection. A wrapper model is proposed in [13] to use a clustering algorithm in evaluating the quality of feature selection.

Recent years have seen many embedded feature selection methods through sparsity regularization, such as the $\ell_{2,1}$-norm of a matrix [11] including supervised embedded methods such as multi-task feature selection [2, 24], spectral feature selection [33] and robust joint $\ell_{2,1}$-Norms [27] and unsupervised embedded feature selection methods such as discriminative unsupervised feature selection [32]. Through sparsity regularization, feature selection can be embedded in the learning process.

We can consider each view as a source, in this perspective, MVFS involves more than one source and it is related to multi-source feature selection [34]. However, multi-view feature selection is different from multi-source feature selection in two ways: (1) multi-source feature selection is designed to select features from the original feature space by integrating multiple sources, while multi-view feature selection select features from different feature spaces simultaneously for all views; and (2) multi-source feature selection ignores the interdependent relations between sources, while multi-view feature selection exploits the relations among views.

6 Conclusion

In this paper, we study a novel problem of unsupervised feature selection for multi-view data, represented by heterogeneous feature spaces. Multiple views are not independent while providing complementary information to each other, presenting both challenges and opportunities to traditional single-view feature selection. A novel unsupervised feature selection framework, MVFS, is proposed for multi-view data, which exploits the relations among views and selects features from all views simultaneously. Experimental results on two multi-view datasets from real-world social media websites show that the proposed framework can improve the performance of both single-view clustering and multi-view clustering.

Social media produces many types of links, containing rich information about social media data. We will investigate how to exploit link information for multi-view feature selection after we have shown the effectiveness of multi-view feature selection.

Acknowledgments

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

References

[1] Lowe, D.G. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[2] A. Argyriou, T. Evgeniou, dand M. Pontil. Multi-task feature learning. *NIPS*, 2007.

[3] S. Bickel and T. Scheffer. Multi-view clustering. *ICDM*, 2004.

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

[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] K. Chaudhuri, S. Kakade, K. Livescu, and K. Sridharan. Multi-view clustering via canonical correlation analysis. *ICML*, 2009.

[7] J. Chen, Z. Zhao, J. Ye, and H. Liu. Nonlinear adaptive distance metric learning for clustering. *KDD*, 2007.

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

[10] C. Ding, T. Li, and M. Jordan. Convex and semi-nonnegative matrix factorizations. *TPAMI*, 2010.

[11] C. Ding, D. Zhou, X. He, and H. Zha. R 1-pca: rotational invariant l 1-norm principal component analysis for robust subspace factorization. *ICML*, 2006.

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

[13] J. Dy and C. Brodley. Feature selection for unsupervised learning. *JMLR*, 2004.

[14] J. G. Dy and C. E. Brodley. Feature subset selection and order identification for unsupervised learning. *ICML*, 2000.

[15] J. G. Dy and C. E. Brodley. Visualization and interactive feature selection for unsupervised data. *KDD*, 2000.
[16] 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.
[17] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 2002.
[18] X. Hu, L. Tang, J. Tang and H. Liu, "Exploiting Social Relations for Sentiment Analysis in Microblogging," in *WSDM*, 2013.
[19] M. Hall. Correlation-based feature selection for discrete and numeric class machine learning. *ICML*, 2000.
[20] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. *NIPS*, 2006.
[21] Y. Kim, W. Street, and F. Menczer. Feature selection for unsupervised learning via evolutionary search. *KDD*, 2000.
[22] H. Liu and H. Motoda. *Computational methods of feature selection*. Chapman & Hall, 2008.
[23] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *TKDE*, 2005.
[24] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient l 2, 1-norm minimization. *UAI*, 2009.
[25] J. Tang, H. Gao, X. Hu and H. Liu, "Exploiting Homophily Effect for Trust Prediction," in *WSDM*,2013.
[26] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 2007.
[27] F. Nie, H. Huang, X. Cai, and C. Ding. Efficient and robust feature selection via joint l21-norms minimization. NIPS, 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] J. Tang and H. Liu. Feature selection with linked data in social media. *SDM*, 2012.
[30] J. Tang and H. Liu. Unsupervised feature selection for linked social media data. *KDD*, 2012.
[31] L. Wolf and A. Shashua. Feature selection for unsupervised and supervised inference: the emergence of sparsity in a weighted-based approach. *JMLR*, 2005.
[32] Y. Yang, H. Shen, Z. Ma, Z. Huang, and X. Zhou. L21-norm regularized discriminative feature selection for unsupervised learning. *IJCAI*, 2011.
[33] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. *ICML*, 2007.
[34] Z. Zhao and H. Liu. Multi-source feature selection via geometry-dependent covariance analysis. *JMLR, Workshop and Conference Proceedings*, 2010.

Appendix

6.1 Proof of Theorem 3.2

Proof. $H(\mathbf{Z})$ can be rewritten as,

$$H(\mathbf{Z}) = Tr\big(\mathbf{Z}^\top \mathbf{M}^+ \mathbf{Z} + \sum_{i=1}^{m} \gamma_i (2\mathbf{A}_i^- \mathbf{Z}^\top + \mathbf{Z}^\top \mathbf{Z}) + \Gamma^+ \mathbf{Z}^\top \mathbf{Z}\big) - Tr\big(\mathbf{Z}^\top \mathbf{M}^- \mathbf{Z} + \sum_{i=1}^{m} 2\gamma_i \mathbf{A}_i^+ \mathbf{Z}^\top + \Gamma^- \mathbf{Z}^\top \mathbf{Z}\big)$$

Then we show that the function $h(\mathbf{Z}, \tilde{\mathbf{Z}})$ is an auxiliary function of $H(\mathbf{Z})$. By the following inequality,

$$2a \le \frac{a^2 + b^2}{b}, \ \forall\ a \ge 0,\ b \ge 0 \tag{6.11}$$

then,

$$\begin{aligned} Tr(\sum_i 2\mathbf{A}_i^- \mathbf{Z}^\top) &= \sum_{ijl} 2\mathbf{A}_i^-(j,l)\mathbf{Z}(j,l) \\ &\le \sum_{ijl} \big(\mathbf{A}_i^-(j,l)\frac{\mathbf{Z}^2(j,l) + \tilde{\mathbf{Z}}^2(j,l)}{\tilde{\mathbf{Z}}(j,l)}\big). \end{aligned} \tag{6.12}$$

It is easy to verify that,

$$\begin{aligned} &Tr\big(\mathbf{Z}^\top \mathbf{M}^+ \mathbf{Z} + \sum_{i=1}^{m} \gamma_i \mathbf{Z}^\top \mathbf{Z} + \Gamma^+ \mathbf{Z}^\top \mathbf{Z}\big) \\ &\le \sum_{ijl} \gamma_i \frac{\tilde{\mathbf{Z}}(j,l)\mathbf{Z}^2(j,l)}{\tilde{\mathbf{Z}}(j,l)} + \sum_{jl} \big(\frac{(\tilde{\mathbf{Z}}\Gamma^+)(j,l)\mathbf{Z}^2(j,l)}{\tilde{\mathbf{Z}}(j,l)} \\ &\quad + \frac{(\mathbf{M}^+\tilde{\mathbf{Z}})(j,l)\mathbf{Z}^2(j,l)}{\tilde{\mathbf{Z}}(j,l)}\big) \end{aligned} \tag{6.13}$$

Due to $z \ge 1 + \log z,\ z \ge 0$, we get,

$$\begin{aligned} &- Tr\big(\mathbf{Z}^\top \mathbf{M}^- \mathbf{Z} + \sum_{i=1}^{m} 2\gamma_i \mathbf{A}_i^+ \mathbf{Z}^\top + \Gamma^- \mathbf{Z}^\top \mathbf{Z}\big) \le \\ &- \Big(\sum_{jl} \big(\sum_i 2\gamma_i \mathbf{A}_i(j,l)\big)\tilde{\mathbf{Z}}(j,l)\big(1 + \log \frac{\mathbf{Z}(j,l)}{\tilde{\mathbf{Z}}(j,l)}\big) \\ &+ \sum_{ijl} \Gamma^-(j,l)\tilde{\mathbf{Z}}(i,j)\tilde{\mathbf{Z}}(i,k)\big(1 + \log \frac{\mathbf{Z}(i,j)\mathbf{Z}(i,l)}{\tilde{\mathbf{Z}}(i,j)\tilde{\mathbf{Z}}(i,k)}\big) \\ &+ \sum_{ijl} \mathbf{M}^-(j,l)\tilde{\mathbf{Z}}(j,i)\tilde{\mathbf{Z}}(k,i)\big(1 + \log \frac{\mathbf{Z}(j,k)\mathbf{Z}(i,l)}{\tilde{\mathbf{Z}}(j,i)\tilde{\mathbf{Z}}(j,l)}\big)\Big). \end{aligned} \tag{6.14}$$

By summing over Eqs. (6.12), (6.13) and (6.14), we can get that $h(\mathbf{Z}, \tilde{\mathbf{Z}}) \ge H(\mathbf{Z})$ and it is also easy to verify that $h(\mathbf{Z}, \mathbf{Z}) = H(\mathbf{Z})$. Thus $h(\mathbf{Z}, \tilde{\mathbf{Z}})$ is an auxiliary function of $H(\mathbf{Z})$.

It is easy to verify that the Hessian matrix of $h(\mathbf{Z}, \tilde{\mathbf{Z}})$ is a diagonal matrix with positive diagonal elements, i.e., a positive definition matrix thus $h(\mathbf{Z}, \tilde{\mathbf{Z}})$ is a convex function and set the derivative to zero, yielding the updating rule in Eq. (3.10), which completes the proof.