2016 IEEE 16th International Conference on Data Mining

Robust Multi-View Feature Selection

Hongfu Liu[1], Haiyi Mao[2] and Yun Fu[1,2]
[1]*Department of Electrical and Computer Engineering, Northeastern University, Boston*
[2]*College of Computer and Information Science, Northeastern University, Boston*
{*liu.hongf, mao.hai*}*@husky.neu.edu, yunfu@ece.neu.edu*

***Abstract*—High-throughput technologies have enabled us to rapidly accumulate a wealth of diverse data types. These multi-view data contain much more information to uncover the cluster structure than single-view data, which draws raising attention in data mining and machine learning areas. On one hand, many features are extracted to provide enough information for better representations; on the other hand, such abundant features might result in noisy, redundant and irrelevant information, which harms the performance of the learning algorithms. In this paper, we focus on a new topic, multi-view unsupervised feature selection, which aims to discover the discriminative features in each view for better explanation and representation. Although there are some exploratory studies along this direction, most of them employ the traditional feature selection by putting the features in different views together and fail to evaluate the performance in the multi-view setting. The features selected in this way are difficult to explain due to the meaning of different views, which disobeys the goal of feature selection as well. In light of this, we intend to give a correct understanding of multi-view feature selection. Different from the existing work, which either incorrectly concatenates the features from different views, or takes huge time complexity to learn the pseudo labels, we propose a novel algorithm, Robust Multi-view Feature Selection (RMFS), which applies robust multi-view K-means to obtain the robust and high quality pseudo labels for sparse feature selection in an efficient way. Nontrivially we give the solution by taking the derivatives and further provide a K-means-like optimization to update several variables in a unified framework with the convergence guarantee. We demonstrate extensive experiments on three real-world multi-view data sets, which illustrate the effectiveness and efficiency of RMFS in terms of both single-view and multi-view evaluations by a large margin.**

***Keywords*-Multi-view Learning; Feature Selection; Robust Clustering**

I. Introduction

Nowadays high dimensional data are ubiquitous in many areas such as text, images, speeches and videos, etc. Many features are extracted to provide enough information for better representations; on the other hand, such abundant features might result in noisy, redundant and irrelevant information, which harms the performance of the learning algorithms. It is very appealing to apply partial features to achieve better performance, which is the goal of feature selection. Especially, many real-world data sets have multiple representations with heterogeneous views [1], [35], [5], [36], [7], [8]. For example, images can be presented in gray level and Fourier coefficient, the news might be raised by different media and the literary works might have multiple translations in different languages. These multi-view data provide much richer information to uncover the intrinsic structure, which has been widely recognized that the multi-view learning reduces the noise, improves statistical significance and obtains more refined and higher-level information [31], [29], [12]. Therefore, these multi-view data also provide new chances for effective feature selection.

Feature selection aims to employ partial original features for the certain task, which is a widely used technique in data mining and machine learning areas [13], [24], [23], [37]. It is important to note that different from feature transformation, such as well-known PCA [18] and Deep Learning [9], which generate new features via linear or non-linear transformation, feature selection only applies partial original features for the learning tasks. Clearly, features after selection provide more interpretation, which is widely used in various applications [16], such as gene expression analysis [25], text mining [11] and image processing [2]. According to the availability of labels, the existing algorithms on this topic can roughly be divided into supervised fashion and unsupervised fashion.

Usually supervised feature selection applies label information to guide the feature selection process [26]; when it comes to unsupervised feature selection, the main challenge is to find appropriate evaluation criteria instead of labels. Such criteria are conductive to explore the intrinsic cluster structure and usually pseudo labels are learnt to guide the feature selection in a supervised fashion [3], [34], [19].

Unfortunately, most of the tradition unsupervised feature selection algorithms can only handle the single-view data and fail to evaluate the performance in the multi-view setting. Nowadays data are gathered from different representations so that multi-data becomes a new research point. Multi-view data provide much more information to uncover the hidden cluster structure than single-view data. It is highly risky to put the features from multi-view together and apply the single-view feature selection methods because the feature spaces and scales are different in each view, furthermore the selected features by this way are difficult to interpret and further analyse, which disobeys the goal of feature selection [10], [15]. In [10], they used the neighbourhood matrices from different views for pseudo labels; [15] incorporated discriminative analysis to preserve the clus-

2374-8486/16 $31.00 © 2016 IEEE
DOI 10.1109/ICDM.2016.37

ter structure. Incorrectly, both of them selected from the concatenating features. [30] employed multi-kernel spectral analysis for pseudo labels and applied it on each single view features. However, how to obtain the robust pseudo labels in an efficient way still remains a big challenge.

In this paper, we propose the Robust Multi-view Feature Selection (RMFS) algorithm to handle the above challenge. It is crucial to obtain high quality pseudo labels to guide the process of unsupervised feature selection [21], [20], [22]. Compared with existing multi-view unsupervised feature selection methods, RMFS not only provides robust and high quality pseudo labels by robust multi-view K-means for feature selection, but also can be solved in an efficient way. Besides, we jointly generate pseudo labels and learn the feature selection in a one-step framework. Then we give a solution by taking the derivative of each unknown variables one by one; further nontrivially by introducing an augmented matrix, a K-means-like optimization solution is designed to simultaneously update several variables in a neatly mathematical way. Recall that most of unsupervised feature selection methods need eigenvector decomposition, which requires $O(n^3)$ for the pseudo labels, here n is the number of data points and becomes struggled to handle large-scale data sets. How to efficiently conduct feature selection on multi-view data is one of our motivations. Our contributions are highlighted in the following three folds:

- We propose a novel Robust Multi-view Feature Selection approach, which provides robust and high quality pseudo labels from multi-view learning to guide the feature selection process and has much lower time complexity than existing methods.
- An efficient algorithm is designed to handle the non-smooth non-convex loss function with convergence guarantee. Further, a K-means-like optimization solution is designed to update several variables in a unified framework.
- Experimental results demonstrate the superior results of RMFS compared with several state-of-the-art methods on both single-view and multi-view evaluations.

The rest of this paper is organized as follows. In Section II, we discuss about related work in terms of unsupervised feature selection and multi-view feature selection. We demonstrate the motivation, the problem we address and the objective function in Section III. Following this, two solutions are provided in Section IV. One is by taking the derivatives, the other is by a K-means-like optimization. Extensive experiments are demonstrated in Section V. We conclude this paper in Section VI.

II. Related Work

Here we illustrate the related work in terms of unsupervised feature selection and multi-view feature selection and highlight the differences between the existing works and ours.

A. Unsupervised Feature Selection

For unsupervised feature selection, the main challenge is to find an appropriate evaluation criteria or high quality pseudo labels instead of true labels. Such criteria are conductive to explore the intrinsic cluster structure and usually pseudo labels are learnt to guide the feature selection in a supervised fashion. By this means, unsupervised feature selection can roughly be divided in three categories: filter, wrapper, embedded approaches. Filter algorithms make use of the proxy measurement to give a score to each feature [17], [14]; wrapper methods incorporate the feature selection and the learning algorithm in a unified framework [38], [14], [28]; embedded methods formalize the feature selection as part of learning objective [17], [6]. Nowadays, the way to learn pseudo labels learnt for feature selection is becoming more and more popular. For instance, MCFS [3] employed spectral analysis and spare regression to select discriminative features, UDFS [34] and NDFS [19] jointly learned the pseudo labels and $\ell_{2,1}$-norm regularization in a unified framework based on spectral learning. Recently, Liu et al. proposed a consensus guided framework for feature selection, which employed consensus clustering to generate pseudo labels for feature selection [21].

B. Multi-view Feature Selection

Nowadays, data are collected by multi-sensors and have several representations, which make multi-view learning a hot research point and multi-view feature selection catches raising attention. Different from single-view data, multi-view date provide much more information to uncover the intrinsic cluster structure. It is crucial to obtain high quality pseudo labels as the feature selection criteria. [10] calculated the neighborhood matrices from each view, then summed them together for the pseudo labels; [15] incorporated discriminative analysis to preserve the cluster structure; [30] employed multi-view spectral clustering and applied the pseudo labels to select features for each view. Unfortunately, [10] and [15] incorrectly concatenated the features from each view and employed the tradition single-view feature selection techniques; by this means, the selected features are not in the same feature space so that the following clustering task on the selected features is meaningless.

Different from existing work, we focus on unsupervised multi-view feature selection and aim to provide a proper understanding for this scenario. Generally speaking we employ the pseudo labels derived from efficient and robust multi-view clustering to supervise the feature selection process and evaluate the performance on the single-view and multi-view settings.

III. Robust Multi-View Feature Selection

In this section, we first illustrate the notation, and then discuss about our motivation. Finally the objective function of Robust Multi-view Feature Selection is given.

Table I
DEFINITION OF NOTATIONS

Notations	Domain	Description
n	$\mathbb{Z}$	#Instance
r	$\mathbb{Z}$	#View
$m^{(v)}$	$\mathbb{Z}$	#Feature in the v-th view
K	$\mathbb{Z}$	#Class
$\mathbf{X}^{(v)}$	$\mathbb{R}^{n\times m^{(v)}}$	Data matrix in the v-th view
$\mathbf{H}$	$\{0,1\}^{n\times K}$	Indicator matrix
$\mathbf{W}^{(v)}$	$\mathbb{R}^{m^{(v)}\times K}$	Feature selection matrix for the v-th view
$\mathbf{G}^{(v)}$	$\mathbb{R}^{K\times m^{(v)}}$	Centroid matrix for the v-th view
$\mathbf{C}^{(v)}$	$\mathbb{R}^{K\times K}$	Alignment matrix for the v-th view

A. Notation

In this paper, we use bold uppercase and lowercase characters to denote matrices and vectors, respectively. For a matrix, $\mathbf{A} \in \mathbb{R}^{n\times m}$, $\mathbf{A}_i$ represents the i-th row of $\mathbf{A}$, A_{ij} denotes the (i,j)-th element of $\mathbf{A}$. $\mathbf{A}^\top$, $\mathbf{A}^{-1}$ and $\mathbf{A}^+$ stand for the transpose, the inverse and the pseudo inverse of a matrix $\mathbf{A}$, respectively. $||\mathbf{A}||_{\mathrm{F}} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}^2}$ is the Frobenius norm and its $\ell_{2,1}$ norm is defined as $||\mathbf{A}||_{2,1} = \sum_{i=1}^{n}\sqrt{\sum_{j=1}^{m} A_{ij}^2}$, and $\mathrm{tr}(\cdot)$ is the trace of a squared matrix. $\mathbf{I}$ is the identity matrix. More definitions of variables are reported in Table I.

B. Motivation

With the rapid development of techniques, it becomes easy to collect the data from different aspects or views. These multi-view data provide much richer information to uncover the intrinsic structure, which has been widely recognized that the multi-view learning reduces the noise, improves statistical significance and obtains more refined and higher-level information [31], [29], [12]. On one hand, many features are extracted to provide enough information for better representations; on the other hand, such abundant features might result in noisy, redundant and irrelevant information, which harms the performance of the learning algorithms. It is very appealing to apply partial features to achieve the same or better performance, which is the goal of feature selection [17], [14]. Although there are some exploratory studies in the multi-view feature selection, most of them put all the features together and apply the single-view feature selection methods [34], [19], [3]. This is of high risk because the feature spaces from different views are different and the selected features by such means are difficult to explain, which disobeys the goal of feature selection. Therefore, one of our motivations is to provide a correct understanding for multi-view feature selection and evaluate the performance in both the single-view and multi-view settings.

C. Objective Function

It has been widely recognized that a sparse projection with pseudo labels is successful to supervise the process of the unsupervised feature selection. Thus, the pseudo labels highly determine the quality of the selected features. Nowadays, many real-world datasets have multiple representations with heterogeneous views, which provide much more complementary and rich information to uncover the intrinsic structure. In light of this, we aim to utilize the heterogeneous information from multiple view to generate high quality pseudo labels for feature selection. A novel algorithm Robust Multi-view Feature Selection (RMFS) is proposed to jointly conduct clustering and sparse learning in an efficient and robust way.

Let $\mathbf{X} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(r)}\}$ be the data with r multiple representations or views, each view $\mathbf{X}^{(v)}$ contains n instances and $m^{(v)}$ features, for $1 \le v \le r$. The objective of RMFS is as follows,

$$\begin{aligned}\min_{\mathbf{H},\mathbf{G},\mathbf{C},\mathbf{W}} \sum_{v=1}^{r} (\alpha^{(v)} ||\mathbf{X}^{(v)} - \mathbf{H}\mathbf{G}^{(v)}||_{2,1} + \\ ||\mathbf{X}^{(v)}\mathbf{W}^{(v)} - \mathbf{H}\mathbf{C}^{(v)}||_{\mathrm{F}}^2 + \beta||\mathbf{W}^{(v)}||_{2,1}),\end{aligned} \tag{1}$$

where $\mathbf{H} \in \{0,1\}^{n\times K}$ is the 1-of-K coding indicator matrix representing the pseudo labels, $\mathbf{G}^{(v)} \in \mathbb{R}^{K\times m^{(v)}}$ is the corresponding centroid matrix of $\mathbf{X}^{(v)}$, $\mathbf{W}^{(v)} \in \mathbb{R}^{n\times K}$ denotes the feature selection matrix for v-th view and $\mathbf{C}^{(v)} \in \mathbb{R}^{K\times K}$ presents the alignment matrix between $\mathbf{X}^{(v)}\mathbf{W}^{(v)}$ and $\mathbf{H}$.

Our objective function consists of three terms. The first part is to obtain the pseudo labels by the multi-view clustering, which is a variant of multi-view K-means with $\ell_{2,1}$ norm to obtain a robust cluster structure [4]; the second term aims to learn feature selection matrices and their corresponding relationship and the last term is the common regularizer. It is worthy to note that we adopt $\ell_{2,1}$ regularization on $\mathbf{W}^{(v)}$ to guarantee that $\mathbf{W}^{(v)}$ is sparse in rows to achieve the goal of feature selection. Multi-view feature selection is a new raising topic, there exist few methods for multi-view feature selection [30]. Different from them, our method has two major differences. One is that we apply the variant of K-means for multi-view clustering, which is more efficient and robust for large-scale datasets instead of multi-view spectral analysis. Another is that we use a crisp indicator matrix $\mathbf{H}$ to represent the cluster structure instead of the soft feature-class mapping matrix, which might suffer from mixed signs and make itself an implicit and distorted representation, and further degrade the performance of feature selection. Since clustering result is orderless, the alignment matrix $\mathbf{C}^{(v)}$ is needed to shuffle the class order in $\mathbf{H}$.

IV. OPTIMIZATION ALGORITHM

The difficulties of solving the proposed objective function in Eq. 1 lies in two points. One is that two terms involve $\ell_{2,1}$-norm, which is non-smooth; another point is that $\mathbf{H}$ is

a binary indicator matrix, rather than a continuous variable. In light of this, we propose a new algorithm to handle the above challenges in an efficient way.

A. Solution by Derivative

Here we take the derivative of each continuous variables alternatively for continuous variables and apply the exhaustive search for the binary variable.

Fixed others, Update $\mathbf{G}^{(v)}$. Only the first term contains $\mathbf{G}^{(v)}$, we have

$$\mathcal{J}^{(v)} = \text{tr}((\mathbf{X}^{(v)} - \mathbf{H}\mathbf{G}^{(v)})^\top \mathbf{D}^{(v)}(\mathbf{X}^{(v)} - \mathbf{H}\mathbf{G}^{(v)})), \quad (2)$$

where $\mathbf{D}^{(v)} \in \mathbb{R}^{n\times n}$ is the diagonal matrix defined as,

$$\mathbf{D}^{(v)}_{ii} = \frac{1}{2||\mathbf{e}^{(v)}_i||}, \quad (3)$$

where $\mathbf{e}^{(v)}_i$ is the i-th row of $\mathbf{X}^{(v)} - \mathbf{H}\mathbf{G}^{(v)}$. Then taking the derivative of $\mathcal{J}^{(v)}$ with $\mathbf{G}^{(v)}$ and setting it to 0, we get

$$\frac{\partial \mathcal{J}^{(v)}}{\partial \mathbf{G}^{(v)}} = -2\mathbf{H}^\top \mathbf{D}^{(v)}\mathbf{X}^{(v)} + 2\mathbf{H}^\top \mathbf{D}^{(v)}\mathbf{H}\mathbf{G}^{(v)} = 0. \quad (4)$$

Then we can update $\mathbf{G}^{(v)}$ as follows,

$$\mathbf{G}^{(v)} = (\mathbf{H}^\top \mathbf{D}^{(v)}\mathbf{H})^{-1}\mathbf{H}^\top \mathbf{D}^{(v)}\mathbf{X}^{(v)}. \quad (5)$$

Fixed others, Update $\mathbf{C}^{(v)}$. Similar to the update rule of $\mathbf{G}^{(v)}$, $\mathbf{C}^{(v)}$ only occurs in the second term. Then we have

$$\mathcal{Q}^{(v)} = \text{tr}((\mathbf{X}^{(v)}\mathbf{W}^{(v)} - \mathbf{H}\mathbf{C}^{(v)})^\top (\mathbf{X}^{(v)}\mathbf{W}^{(v)} - \mathbf{H}\mathbf{C}^{(v)})). \quad (6)$$

Taking the derivative of $\mathcal{Q}^{(v)}$ with $\mathbf{C}^{(v)}$, we get

$$\frac{\partial \mathcal{Q}^{(v)}}{\partial \mathbf{C}^{(v)}} = -2\mathbf{H}\mathbf{X}\mathbf{W}^{(v)} + 2\mathbf{H}^\top \mathbf{H}\mathbf{C}^{(v)}. \quad (7)$$

Setting Eq.7 to 0, we have the update rule of $\mathbf{C}^{(v)}$,

$$\mathbf{C}^{(v)} = (\mathbf{H}^\top \mathbf{H})^{-1}\mathbf{H}\mathbf{X}\mathbf{W}^{(v)}. \quad (8)$$

Fixed others, Update H. $\mathbf{H}$ is the binary indicator matrix. Thus we assign each instance to every cluster and find the label to minimize the objective function.

$$\min_{\mathbf{H}_i} \sum_{v=1}^{r} \alpha^{(v)} ||\mathbf{X}^{(v)}_i - \mathbf{H}_i\mathbf{G}^{(v)}||_2^2 + ||\mathbf{X}^{(v)}_i\mathbf{W}^{(v)} - \mathbf{H}_i\mathbf{C}^{(v)}||_2^2. \quad (9)$$

Since $\mathbf{H}$ is a crisp indictor matrix, we have only one non-zero element in $\mathbf{H}_i$, therefore we do the exhaustive search to find the optimal cluster label.

Fixed others, Update $\mathbf{W}^{(v)}$. Let $\mathcal{L}^{(v)} = ||\mathbf{X}^{(v)}\mathbf{W}^{(v)} - \mathbf{H}\mathbf{C}^{(v)}||_{\text{F}} + \beta||\mathbf{W}^{(v)}||_{2,1}$, we have

$$\frac{\partial \mathcal{L}^{(v)}}{\partial \mathbf{W}^{(v)}} = 2\mathbf{X}^{(v)\top}(\mathbf{X}^{(v)}\mathbf{W}^{(v)} - \mathbf{H}\mathbf{C}^{(v)}) + \beta\mathbf{F}^{(v)}\mathbf{W}^{(v)}, \quad (10)$$

where $\mathbf{F}^{(v)}$ is $\text{diag}(\frac{1}{2||\mathbf{W}^{(v)}_1||_2}, \cdots, \frac{1}{2||\mathbf{W}^{(v)}_{m^{(v)}}||_2})$. When $\frac{\partial \mathcal{L}^{(v)}}{\partial \mathbf{W}^{(v)}} = 0$, we have the update rule for $\mathbf{W}^{(v)}$.

Algorithm 1 Robust Multi-View Feature Selection

Input: $\mathbf{X} = \{\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(r)}\}$: data matrix with r views;
r: number of views;
K: number of clusters;
$\alpha^{(v)}, \beta$: trade-off parameters;
Output: $\mathbf{H}$: indicator matrix;
$\mathbf{G}^{(v)}$: centroid matrix for the v-th view;
$\mathbf{W}^{(v)}$: feature selection matrix for the v-th view;
$\mathbf{C}^{(v)}$: alignment matrix for the v-th view;

Initialize $\mathbf{H}$, $\mathbf{G}^{(v)}$, $\mathbf{W}^{(v)}$ and $\mathbf{C}^{(v)}$;
repeat
  For each view, calculate $\mathbf{D}^{(v)}$ by Eq. 3;
  For each view, fix others, update $\mathbf{G}^{(v)}$ by Eq. 5;
  For each view, fix others, update $\mathbf{C}^{(v)}$ by Eq. 8;
  Fix other, update $\mathbf{H}$ by Eq. 9;
  For each view, fix others, update $\mathbf{W}^{(v)}$ by Eq. 11;
until the objective value in Eq. 1 is unchanged;
return $\mathbf{H}$, $\mathbf{G}^{(v)}$, $\mathbf{W}^{(v)}$ and $\mathbf{C}^{(v)}$;
For v-th view, sort all $m^{(v)}$ features according to $||\mathbf{W}^{(v)}_i||_2$ in descending order and select the certain number of ranked ones.

$$\mathbf{W}^{(v)} = (\mathbf{X}^{(v)\top}\mathbf{X}^{(v)} + \beta\mathbf{F}^{(v)})^{-1}\mathbf{X}^{(v)\top}\mathbf{H}\mathbf{C}^{(v)}. \quad (11)$$

Fixed others, Update $\mathbf{D}^{(v)}$. The update rule for $\mathbf{D}^{(v)}$ is given in Eq. 3.

In sum, we iteratively update these unknown variables and summarize the algorithm for Eq. 1 in Algorithm 1. After obtaining the feature selection matrix $\mathbf{W}$, for each view we sort all $m^{(v)}$ features according to $||W^{(v)}_i||2$ in a descending order and select the certain number of top ones. Since we do not use labels during the training process, clustering is employed for evaluating the performance of selected features.

B. K-means-like Solution

In Algorithm 1, we update the auxiliary matrices $\mathbf{G}$ and $\mathbf{C}$ for each view and update the pseudo labels $\mathbf{H}$. Besides, there are a lot of matrix inverse and multiplication, which is time consuming. Can we update these variables together in a more efficient way?

The answer is positive. Let $\mathbf{A}^{(v)} = (\alpha^{(v)}\mathbf{D}^{(v)})^{1/2}$, where $\mathbf{D}^{(v)}$ is defined in Eq. 3. Then we can rewrite the objective function as follows.

$$\min_{\mathbf{H},\mathbf{G},\mathbf{C},\mathbf{W}} \sum_{v=1}^{r} (\alpha^{(v)}||\mathbf{A}^{(v)}\mathbf{X}^{(v)} - \mathbf{A}^{(v)}\mathbf{H}\mathbf{G}^{(v)}||_{\text{F}}^2 + ||\mathbf{X}^{(v)}\mathbf{W}^{(v)} - \mathbf{H}\mathbf{C}^{(v)}||_{\text{F}}^2 + \beta||\mathbf{W}^{(v)}||_{2,1}). \quad (12)$$

Next we update all the unknown variables except $\mathbf{W}^{(v)}$ in a one-step framework.

Fixed $\mathbf{A}^{(v)}$ and $\mathbf{W}^{(v)}$, Update others. The subproblem is related to the first two terms. Let $\mathcal{Z} = \sum_{i=1}^{r} \mathcal{Z}^{(v)}$, with

$$\begin{aligned}\mathcal{Z}^{(v)} &= ||\mathbf{A}^{(v)}(\mathbf{X}^{(v)} - \mathbf{H}\mathbf{G}^{(v)})||_{\mathrm{F}} + ||\mathbf{X}^{(v)}\mathbf{W}^{(v)} - \mathbf{H}\mathbf{C}^{(v)}||_{\mathrm{F}}^2 \\ &= |||[\mathbf{A}^{(v)}\mathbf{X}^{(v)}\ \mathbf{X}^{(v)}\mathbf{W}^{(v)}] - [\mathbf{A}^{(v)}\ \mathbf{I}]\mathbf{H}[\mathbf{G}^{(v)}\ \mathbf{C}^{(v)}]||_{\mathrm{F}}^2.\end{aligned} \tag{13}$$

Next we build three matrices for each view as follows.

$$\begin{aligned}\mathbf{U}^{(v)} &= [\mathbf{A}^{(v)}\ \mathbf{I}], \\ \mathbf{R}^{(v)} &= [\mathbf{A}^{(v)}\mathbf{X}^{(v)}\ \mathbf{X}^{(v)}\mathbf{W}^{(v)}], \\ \mathbf{V}^{(v)} &= [\mathbf{G}^{(v)}\ \mathbf{C}^{(v)}].\end{aligned} \tag{14}$$

Then we have the concatenated matrices of all r views as $\mathbf{U} = [\mathbf{U}^{(1)}, \cdots, \mathbf{U}^{(r)}]$, $\mathbf{R} = [\mathbf{R}^{(1)}, \cdots, \mathbf{R}^{(r)}]$ and $\mathbf{V} = [\mathbf{V}^{(1)}, \cdots, \mathbf{V}^{(r)}]$. By the following theorem, we provide an efficient way to update these variables in a unified framework.

Theorem 1. *Given the concatenated matrices $\boldsymbol{U}, \boldsymbol{R}$ and $\boldsymbol{V}$ built by Eq. 14, we have the following equivalency:*

$$\min \mathcal{Z} \Leftrightarrow \min_{\boldsymbol{H},\boldsymbol{V}} ||\boldsymbol{U}^{+}\boldsymbol{R} - \boldsymbol{H}\boldsymbol{V}||_{\boldsymbol{F}}^2. \tag{15}$$

Proof: According to Eq. 13 and 14, we have

$$\begin{aligned}\mathcal{Z} &= \sum_{i=1}^{r} \mathcal{Z}^{(r)} \\ &= \sum_{i=1}^{r} |||[\mathbf{A}^{(v)}\mathbf{X}^{(v)}\ \mathbf{X}^{(v)}\mathbf{W}^{(v)}] - [\mathbf{A}^{(v)}\ \mathbf{I}]\mathbf{H}[\mathbf{G}^{(v)}\ \mathbf{C}^{(v)}]||_{\mathrm{F}}^2 \\ &= \sum_{i=1}^{r} ||\mathbf{R}^{(v)} - \mathbf{U}^{(v)}\mathbf{H}\mathbf{V}^{(v)}||_{\mathrm{F}}^2 \\ &= ||\mathbf{R} - \mathbf{U}\mathbf{H}\mathbf{V}||_{\mathbf{F}}^2 = ||\mathbf{U}(\mathbf{U}^{+}\mathbf{R} - \mathbf{H}\mathbf{V})||_{\mathbf{F}}^2.\end{aligned} \tag{16}$$

Since $\mathbf{U}$ is a constant, we complete the proof. ■

Remark 1. *Theorem 1 gives a new insight to update $2r+1$ variables in a unified framework. If we take a close look of Eq. 15, the right side is just the standard K-means, which indicates that we can directly use the simplest clustering algorithm to finish the update. All the information of $\boldsymbol{G}^{(v)}$ and $\boldsymbol{C}^{(v)}$ is summarized in the matrix $\boldsymbol{V}$.*

Remark 2. *The K-means is conducted on the new matrix $\boldsymbol{U}^{+}\boldsymbol{R}$. Here it is worthy to note that $\boldsymbol{U}$ consists of $2r$ diagonal matrices, therefore the pseudo inverse of $\boldsymbol{U}$ takes little time.*

Remark 3. *Compared with the update rules in Algorithm 1, we can see that the pseudo labels $\boldsymbol{H}$ is only updated once with all the information from different views. It indicates that the pseudo labels $\boldsymbol{H}$ is generated or updated in a consensus way.*

Fixed others, Update $\mathbf{W}^{(v)}$. Here we can still use the update rule in Eq. 11 to update $\mathbf{W}^{(v)}$ for each view.

We summarize the alternative solution for Robust Multi-view Feature Selection in Algorithm 2.

Algorithm 2 Robust Multi-View Feature Selection

Input: $\mathbf{X} = \{\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(r)}\}$: data matrix with r views;
r: number of views;
K: number of clusters;
$\alpha^{(v)}, \beta$: trade-off parameters;
Output: $\mathbf{H}$: indicator matrix;
$\mathbf{G}^{(v)}$: centroid matrix for the v-th view;
$\mathbf{W}^{(v)}$: feature selection matrix for the v-th view;
$\mathbf{C}^{(v)}$: alignment matrix for the v-th view;

1: Initialize $\mathbf{H}$, $\mathbf{G}^{(v)}$, $\mathbf{W}^{(v)}$ and $\mathbf{C}^{(v)}$;
2: **repeat**
3: Build the concatenated matrices $\mathbf{U}, \mathbf{R}$ and $\mathbf{V}$ by Eq. 14;
4: Run K-means on $\mathbf{U}^{+}\mathbf{R}$ to obtain $\mathbf{H}$, $\mathbf{G}^{(v)}$ and $\mathbf{C}^{(v)}$;
5: For each view, fix others, update $\mathbf{W}^{(v)}$ by Eq. 11;
6: **until** the objective value in Eq. 1 is unchanged;
7: **return** $\mathbf{H}$, $\mathbf{G}^{(v)}$, $\mathbf{W}^{(v)}$ and $\mathbf{C}^{(v)}$;
8: For v-th view, sort all $m^{(v)}$ features according to $||\mathbf{W}_i^{(v)}||_2$ in descending order and select the certain number of ranked ones.

C. Discussion and Analysis

Finally we will give the convergence study of Algorithm 2 by the following theorem.

Theorem 2. *The objective function value of Eq. 1 continuously decreases by the alternative updating rules in Algorithm 2.*

Proof: In the above solution, we decompose the optimization problem of Eq. 1 into two subproblems. The convergence is guaranteed if both sub problems make the objective function value continuously decrease. One is to update the pseudo labels $\mathbf{H}$ and some other auxiliary variables; the other is to update the feature selection matrix $\mathbf{W}^{(v)}$. Since we transform the first subproblem into a K-means optimization problem, which has the good convergence property, in the following we focus on the second subproblem.

When other variables are fixed, for the v-th view, let $\mathbf{W} = \mathbf{W}^{(v)}$, $\mathbf{F} = \mathbf{F}^{(v)}$ and $\Omega(\mathbf{W}) = ||\mathbf{X}\mathbf{W}^{(v)} - \mathbf{H}\mathbf{C}^{(v)}||_{\mathrm{F}}^2$, in the t-th iteration, we have

$$\begin{aligned}&\mathbf{W}_{t+1} = argmin_{\mathbf{W}}\ \Omega(\mathbf{W}_t) + \beta \mathrm{tr}(\mathbf{W}_t^{\top}\mathbf{F}_t\mathbf{W}_t) \\ &\Rightarrow \Omega(\mathbf{W}_{t+1}) + \beta \mathrm{tr}(\mathbf{W}_{t+1}^{\top}\mathbf{F}_t\mathbf{W}_{t+1}) \leq \Omega(\mathbf{W}_t) + \beta \mathrm{tr}(\mathbf{W}_t^{\top}\mathbf{F}_t\mathbf{W}_t) \\ &\Rightarrow \Omega(\mathbf{W}_{t+1}) + \beta \sum_{i=1}^{m} \frac{||(\mathbf{w}_{t+1})_i||_2^2}{2||(\mathbf{w}_t)_i||_2} \leq \Omega(\mathbf{W}_t) + \beta \sum_{i=1}^{m} \frac{||((\mathbf{w}_t)_i||_2^2}{2||(\mathbf{w}_t)_i||_2} \\ &\Rightarrow \Omega(\mathbf{W}_{t+1}) + \beta||\mathbf{W}_{t+1}||_{2,1} - \beta(||\mathbf{W}_{t+1}||_{2,1} - \sum_{i=1}^{m} \frac{||((\mathbf{w}_{t+1})_i||_2^2}{2||(\mathbf{w}_t)_i||_2}) \\ &\leq \Omega(\mathbf{W}_t) + \beta||\mathbf{W}_t||_{2,1} - \beta(||\mathbf{W}_t||_{2,1} - \sum_{i=1}^{m} \frac{||(\mathbf{w}_t)_i||_2^2}{2||(\mathbf{w}_t)_i||_2}).\end{aligned}$$

According to the lemma in [27], we have

$$||\mathbf{W}_{t+1}||_{2,1} - \sum_{i=1}^{m} \frac{||((\mathbf{w}_{t+1})_i||_2^2}{2||(\mathbf{w}_t)_i||_2} \leq ||\mathbf{W}_t||_{2,1} - \sum_{i=1}^{m} \frac{||(\mathbf{w}_t)_i||_2^2}{2||(\mathbf{w}_t)_i||_2} \tag{17}$$

Then we have

$$\Omega(\mathbf{W}_{t+1}) + \beta||\mathbf{W}_{t+1}||_{2,1} \leq \Omega(\mathbf{W}_t) + \beta||\mathbf{W}_t||_{2,1} \tag{18}$$

Therefore, we prove the objective function value will continuously decrease during the subproblem for updating $\mathbf{W}^{(v)}$. Similarly, the objective function value will also continuously decrease when updating other feature selection matrices. In sum, the objective function value of Eq. 1 continuously decreases by the alternative updating rules in Algorithm 2 and we finish the proof. ■

Next we give the analyses of the computational complexity of RMFS. When updating the pseudo labels, it has the similar time complexity with traditional K-means, $O(nKr(m+Kr))$, where $m = \sum_{i=1}^{r} m^{(v)}$, n is the number of data points; when updating $\mathbf{W}^{(v)}$, it takes $\mathcal{O}((m^{(v)})^3)$. Therefore, the total time cost is $O(InKr(m + Kr) + I\sum_{v=1}^{r}(m^{(v)})^3)$, where I is the number of iteration. Recall that for these methods which apply the spectral analysis for pseudo labels, the eigenvector-decomposition is indispensable. They require $O(n^3)$ for the pseudo labels and struggle to handle large-scale data sets.

Compared to the model in [30], the main difference is how to get the pseudo labels from the multi-view data. They employed the multi-kernel spectral analysis, while in our paper multi-view K-means is used for the consensus partition. The benefits lie in three aspects. One is that the $\ell_{2,1}$-norm makes K-means robust to outliers and provides the high quality pseudo labels, the second point is that eignevector decomposition is replaced by the linear K-means, which dramatically decreases the time complexity and the last point is that we can involve all the variables except the feature selection matrices in a unified K-means optimization framework.

V. Experiments

In this section, we demonstrate extensive experimental results on three widely used multi-view data sets. Besides the comparison with single-view feature selection and multi-view feature selection methods in terms of effectiveness and efficiency, we also show the high performance of RMFS on the multi-view clustering setting. Finally, the convergence study of RMFS also verifies the correctness of Theorem 2.

A. Experimental Setup

Three public multi-view data sets are used for evaluating the proposed method. The details of these data sets can be found in Table II.

Table II
Experimental Data Sets

View	Digits	Movie	MINIST-USPS
1	Pixel(240)	Keywords(1878)	MNIST(256)
2	Fourier(74)	Actors(1398)	USPS (256)
#Instances	2000	617	1630
#Classes	10	17	10

- *Digits*[1] is a handwritten digit dataset from UCI repository. Each data point is represented by grey pixel and Fourier coefficient.
- *Movie*[2] is a dataset has been extracted from IMDb[3] to have two data matrices descibing the same movies. It has been used in co-clustering tasks, the main goal is to find the genre of the movies, combining the information from the two matrices (keywords and actors).
- *MNIST-USPS*[4] is a combination of two data sets MNIST and USPS. Both of them are the grayscale images of 0 through 9 Two data sets are combined together as two views in a multi-view data sets.

Since no label information is used during the training process, we make use of the clustering framework to evaluate the performance in the single-view and multi-view setting. Unsupervised feature selection on multi-view data is a raising topic, so that there are few studies on this area. To our best knowledge, MVFS [30] is the most related competitive algorithm to our setting. The following are several competitive algorithms.

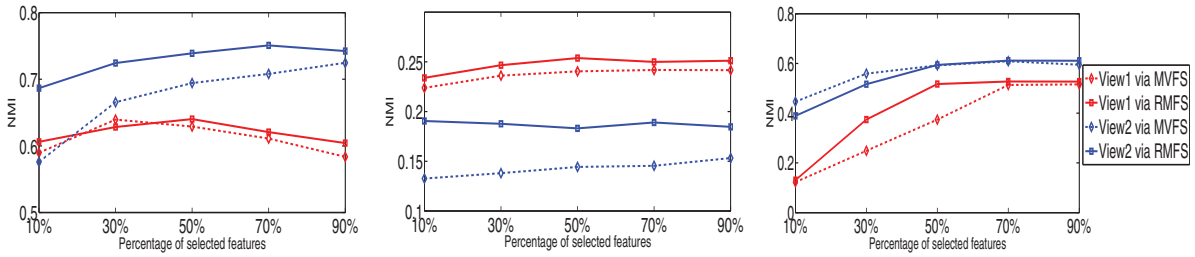
- **MaxVar** ranks the features by their variances and selects the ones with large variances.
- **LS** [14] explores the local manifold structure and picks up the features which have the most consistency with Gaussian Laplacian matrix.
- **MCFS** [3] employs the spectral analysis and spare regression for the pseudo labels and uses the pseudo labels to rank the features.
- **UDFS** [34] builds a joint framework to learn the discriminative analysis and feature selection together.
- **NDFS** [19] selects the discriminative features by applying the Nonnegative spectral analysis with $\ell_{2,1}$-norm regularization.
- **MVFS** [30] generates pseudo labels by multi-view spectral analysis to guide the feature selection for each view.
- **RMFS**. Our proposed method based on robust multi-view learning.

Following the setting of other works [3], [14], [19], [34], we set the number of neighborhoods to be 5 for LS, MCFA, NDFS when building the Laplacian graph. The

[1] http://archive.ics.uci.edu/ml/datasets.html
[2] http://lig-membres.imag.fr/grimal/data.html
[3] http://www.imdb.org
[4] http://www.cs.nyu.edu/~roweis/data.html

Table III
PERFORMANCE OF DIFFERENT ALGORITHMS ON *Digits* MEASURED BY ACCURACY AND NMI.

View	Percentage	Accuracy						
		MaxVar	LS	MCFS	UDFS	NDFS	MVFS	RMFS
View1	0.1	0.5641 ± 0.0579	0.6548 ± 0.0227	0.6259 ± 0.0333	0.5703 ± 0.0329	0.6458 ± 0.0279	0.6519 ± 0.0383	**0.6711 ± 0.0337**
	0.3	0.6040 ± 0.0475	0.6307 ± 0.0480	0.6113 ± 0.0402	0.5710 ± 0.0401	0.5825 ± 0.0588	0.6416 ± 0.0608	**0.6689 ± 0.0337**
	0.5	0.5759 ± 0.0567	0.6096 ± 0.0503	0.6366 ± 0.0490	0.5349 ± 0.0533	0.5597 ± 0.0480	0.6300 ± 0.0406	**0.6548 ± 0.0563**
	0.7	0.5883 ± 0.0450	0.5803 ± 0.0536	0.6220 ± 0.0370	0.5759 ± 0.0455	0.5184 ± 0.0499	0.6120 ± 0.0522	**0.6344 ± 0.0399**
	0.9	0.5709 ± 0.0562	**0.6024 ± 0.0520**	0.5611 ± 0.0574	0.5776 ± 0.0597	0.5245 ± 0.0497	0.5738 ± 0.0494	0.5957 ± 0.0181
View2	0.1	0.5891 ± 0.0315	0.5224 ± 0.0495	0.6095 ± 0.0359	0.4008 ± 0.0209	0.3911 ± 0.0243	0.5503 ± 0.0395	**0.6361 ± 0.0815**
	0.3	0.6991 ± 0.0447	0.6053 ± 0.0413	0.6756 ± 0.0458	0.4855 ± 0.0248	0.5499 ± 0.0486	0.6270 ± 0.0487	**0.6863 ± 0.0412**
	0.5	0.7228 ± 0.0670	0.6615 ± 0.0380	0.7266 ± 0.0801	0.5549 ± 0.0378	0.6351 ± 0.0496	0.6611 ± 0.0650	**0.7739 ± 0.0723**
	0.7	0.7276 ± 0.0758	0.7068 ± 0.0702	0.7074 ± 0.0731	0.6107 ± 0.0423	0.6914 ± 0.0537	0.6822 ± 0.0689	**0.7590 ± 0.0494**
	0.9	0.7079 ± 0.0795	0.6840 ± 0.0596	0.6781 ± 0.0614	0.6291 ± 0.0306	0.6767 ± 0.0610	0.7131 ± 0.0670	**0.7788 ± 0.1091**
		NMI						
View1	0.1	0.5683 ± 0.0237	0.4949 ± 0.0143	0.5990 ± 0.0255	0.5552 ± 0.0160	0.5970 ± 0.0217	0.5897 ± 0.0138	**0.6061 ± 0.0163**
	0.3	0.5873 ± 0.0235	0.4864 ± 0.0327	0.6113 ± 0.0194	0.5570 ± 0.0195	0.5726 ± 0.0277	**0.6394 ± 0.0165**	0.6283 ± 0.0095
	0.5	0.5632 ± 0.0230	0.4865 ± 0.0382	0.6258 ± 0.0212	0.5558 ± 0.0252	0.5619 ± 0.0221	0.6290 ± 0.0228	**0.6402 ± 0.0416**
	0.7	0.5681 ± 0.0162	0.4586 ± 0.0334	0.5958 ± 0.0132	0.5754 ± 0.0188	0.5287 ± 0.0172	0.6110 ± 0.0242	**0.6207 ± 0.0304**
	0.9	0.5609 ± 0.0229	0.4647 ± 0.0320	0.5690 ± 0.0255	0.5802 ± 0.0254	0.5317 ± 0.0168	0.5836 ± 0.0253	**0.6042 ± 0.0318**
View2	0.1	0.5551 ± 0.0124	0.3767 ± 0.0312	0.5998 ± 0.0108	0.3915 ± 0.0132	0.3953 ± 0.0102	0.5759 ± 0.0269	**0.6866 ± 0.0230**
	0.3	0.6577 ± 0.0205	0.4823 ± 0.0300	0.6801 ± 0.0254	0.4871 ± 0.0099	0.5537 ± 0.0187	0.6655 ± 0.0272	**0.7243 ± 0.0296**
	0.5	0.6846 ± 0.0328	0.5457 ± 0.0340	0.7169 ± 0.0362	0.5578 ± 0.0141	0.6438 ± 0.0179	0.6943 ± 0.0325	**0.7389 ± 0.0350**
	0.7	0.7105 ± 0.0363	0.5998 ± 0.0618	0.7177 ± 0.0365	0.6394 ± 0.0226	0.7046 ± 0.0270	0.7080 ± 0.0364	**0.7508 ± 0.0591**
	0.9	0.7013 ± 0.0412	0.5823 ± 0.0507	0.7069 ± 0.0333	0.6409 ± 0.0195	0.6957 ± 0.0307	0.7247 ± 0.0372	**0.7423 ± 0.0362**



(a) *Digits* (b) *Movie* (c) *MNIST-USPS*

Figure 1. Performance comparison between MVFS and RMFS on single-view evaluation with different ratios.

sparse parameter is set to 0.01 for those methods employing pseudo label to guide the feature selection. For MVFS and RMFS, the weight of each view is equal for simplicity. The numbers of selected feature ratio vary from 10% to 90% with 20% intervals. Since no label information is involved during the training process, we evaluate the performance in the clustering scenario. Specially, we employ *k-means* by MATLAB with the true cluster number on the selected features and evaluate the performance by the external measurement. For each algorithm, we run 50 times and report the average result and standard deviation.

Two widely used external cluster validity metrics are used to fully evaluation the performance, Accuracy and Normalized Mutual Information (NMI) [32]. Accuracy is a measure derived from classification, which needs the mapping between the obtained partition and ground truth. NMI measures the mutual dependence between obtained cluster labels and ground truth, followed by a normalization operation to make sure NMI range from 0 to 1. Both of them are positive measurements, which indicate that the larger value stands for better performance. The computation of these two metrics are as follows:

$$Accuracy = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n}, \tag{19}$$

where s_i and r_i are the predicted and true labels for the i-th data point, $\delta(x, y)$ equals one if $x = y$ and equals zero otherwise, and $map(r_i)$ is the permutation mapping function that best aligns the clusters in the learnt partition with ground truth.

$$NMI = \frac{\sum_{i,j} n_{ij} \log \frac{n \cdot n_{ij}}{n_{i+} \cdot n_{+j}}}{\sqrt{(\sum_i n_{i+} \log \frac{n_{i+}}{n})(\sum_j n_{j+} \log \frac{n_{+j}}{n})}}, \tag{20}$$

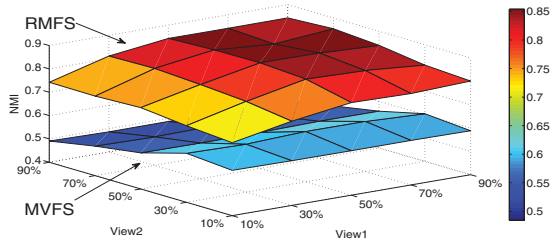
where n_{ij} denotes the co-occurrence instance number in the i-th cluster of the ground truth and j-th cluster of the obtained partition and n_{i+} and n_{+j} are the cluster size of the i-th cluster of the ground truth and j-th cluster of the obtained partitions, respectively.

B. Performance and Analysis

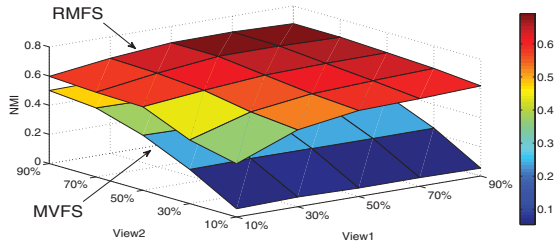
Table III, IV and V show the performance of several feature selection methods in the single-view setting on *Digits*, *Moive* and *MNIST-USPS*. The best results are highlighted by bold fonts. Generally speaking, the results demonstrate two

Table IV
PERFORMANCE OF DIFFERENT ALGORITHMS ON *Movie* MEASURED BY ACCURACY AND NMI.

View	Percentage	Accuracy						
		MaxVar	LS	MCFS	UDFS	NDFS	MVFS	RMFS
View1	0.1	0.2184 ± 0.0152	0.2073 ± 0.0139	0.1995 ± 0.0137	0.1965 ± 0.0131	0.2169 ± 0.0148	0.1583 ± 0.0124	**0.2257 ± 0.0157**
	0.3	0.2081 ± 0.0188	0.2170 ± 0.0106	0.2146 ± 0.0134	0.2035 ± 0.0118	0.2102 ± 0.0094	0.1884 ± 0.0190	**0.2227 ± 0.0091**
	0.5	0.2115 ± 0.0103	0.2062 ± 0.013	0.2084 ± 0.0089	0.2103 ± 0.0074	0.2119 ± 0.0086	0.1875 ± 0.0174	**0.2146 ± 0.0077**
	0.7	0.2176 ± 0.0104	0.1967 ± 0.0216	0.2093 ± 0.0095	0.2135 ± 0.0066	0.2107 ± 0.0074	0.1950 ± 0.0131	**0.2173 ± 0.0094**
	0.9	0.2111 ± 0.0121	0.2110 ± 0.0143	0.2115 ± 0.0111	0.2114 ± 0.0062	**0.2131 ± 0.0120**	0.2043 ± 0.0099	0.2118 ± 0.0073
View2	0.1	0.1391 ± 0.0075	0.1390 ± 0.0081	0.1414 ± 0.0054	0.1182 ± 0.0067	0.1144 ± 0.0097	0.1076 ± 0.0051	**0.1504 ± 0.0097**
	0.3	0.1421 ± 0.0056	0.1437 ± 0.0050	0.1559 ± 0.0048	0.1233 ± 0.0049	0.1220 ± 0.0056	0.1136 ± 0.0072	**0.1504 ± 0.0095**
	0.5	0.1361 ± 0.0044	0.1349 ± 0.0047	**0.1498 ± 0.0106**	0.1306 ± 0.0074	0.1295 ± 0.0071	0.1184 ± 0.0071	0.1462 ± 0.0106
	0.7	0.1361 ± 0.0083	0.1378 ± 0.0066	0.1453 ± 0.0080	0.1310 ± 0.0038	0.1334 ± 0.0064	0.1216 ± 0.0082	**0.1467 ± 0.0117**
	0.9	0.1341 ± 0.0103	0.1374 ± 0.0075	0.1397 ± 0.0055	0.1340 ± 0.0077	0.1380 ± 0.0099	0.1265 ± 0.0086	**0.1418 ± 0.0100**
		NMI						
View1	0.1	0.2296 ± 0.0108	0.2026 ± 0.0171	0.2080 ± 0.0151	0.2061 ± 0.0137	0.2230 ± 0.0170	0.2238 ± 0.0205	**0.2339 ± 0.0149**
	0.3	0.2392 ± 0.0133	0.2191 ± 0.0123	0.2235 ± 0.0130	0.2209 ± 0.0155	0.2302 ± 0.0104	0.2361 ± 0.0139	**0.2466 ± 0.0136**
	0.5	0.2404 ± 0.0124	0.2168 ± 0.0129	0.2326 ± 0.0089	0.2318 ± 0.0093	0.2413 ± 0.0119	0.2404 ± 0.0112	**0.2538 ± 0.0168**
	0.7	0.2412 ± 0.0090	0.2112 ± 0.0200	0.2388 ± 0.0143	0.2402 ± 0.0112	0.2407 ± 0.0118	0.2418 ± 0.0123	**0.2499 ± 0.0097**
	0.9	0.2508 ± 0.0138	0.2233 ± 0.0166	0.2448 ± 0.0106	0.2460 ± 0.0085	0.2446 ± 0.0114	0.2416 ± 0.0111	**0.2511 ± 0.0117**
View2	0.1	0.1708 ± 0.0097	0.1811 ± 0.0089	0.1665 ± 0.0068	0.1344 ± 0.0051	0.1388 ± 0.0138	0.1326 ± 0.0088	**0.1905 ± 0.0189**
	0.3	0.1789 ± 0.0119	0.1789 ± 0.0134	0.1734 ± 0.0087	0.1391 ± 0.0061	0.1550 ± 0.0080	0.1380 ± 0.0094	**0.1877 ± 0.0129**
	0.5	0.1648 ± 0.0121	0.1791 ± 0.0072	0.1677 ± 0.0098	0.1476 ± 0.0078	0.1645 ± 0.0075	0.1443 ± 0.0099	**0.1831 ± 0.0136**
	0.7	0.1647 ± 0.0118	0.1821 ± 0.0121	0.1743 ± 0.0114	0.1763 ± 0.0132	0.1687 ± 0.0069	0.1454 ± 0.0092	**0.1890 ± 0.0172**
	0.9	0.1651 ± 0.0167	0.1797 ± 0.0097	0.1772 ± 0.0084	0.1633 ± 0.0044	0.1755 ± 0.0112	0.1532 ± 0.0107	**0.1846 ± 0.0175**



(a) *Digits*



(b) *MNIST-USPS*

Figure 2. Performance comparison between MVFS and RMFS on multi-view clustering evaluation.

representative scenarios between single-view and multi-view feature selection. On *Digits*, multi-view feature selection methods including MVFS and our proposed RMFS almost outperform all other single-view feature selection methods by a large margin within all selected feature ratios. For example, RMFS exceeds NDFS by 8% and 14% in the 30% setting of View-1 and View-2, respectively and RMFS exceeds UDFS by 12% and 20% in the 50% setting of View-1 and View-2 in terms of NMI. That indicates the multi-view data provide rich information so that high quality pseudo labels derived from multi-view data give better guidance for feature selection. On *Moive*, especially on *MNIST-USPS*, multi-view feature selection methods struggle to compete with single-view methods. This is mainly because the structures hidden in each view are inconsistent to each other, which leads the multi-view methods to learn a moderate structure heavily different from the ones learned from single-view data. However, with more selected features, the performance of multi-view feature selection methods consistently increases. Although our method RMFS performs not well on single-view evaluation on *MNIST-USPS*, the performance will boost in the multi-view evaluation (we will show that later).

Compared with the multi-view feature selection method MVFS, Figure 1 shows the comparative results between MVFS and RMFS on three data sets. It is obvious that RMFS substantially exceeds MVFS in all the cases except the one with 30% features on the View-1 on *Digits*. For instance, RMFV has over 7%, 8%, 10% improvements over MVFS in the View-2 on *Digits* with 30% selected features, in View-1 on *Movie* with 10% selected features and in View-1 on *MNIST-USPS* with 50% selected features. This reveals that the robust K-means term is more effective than multi-kernel spectral in learning pseudo labels. Other than the robust cluster structure, our method also has the linear time complexity during the subproblem updating $\mathbf{H}$; however, eignevector decomposition is indispensable, which takes $O(n^3)$ time complexity and prevents itself to handel large-scale data sets.

Further, we evaluate the performance in multi-view setting. Figure 2 shows multi-view clustering results based on the portfolios of different feature ratios from two views based on the multi-view clustering method [4]. We can see that our proposed RMVC beats MVFS in all the combination with different feature ratios from two views. It is worthy to note that on *Digits*, the worst result of RMVC (NMI = 0.7129, with 10% features from View-1 and 10% features

Table V
PERFORMANCE OF DIFFERENT ALGORITHMS ON *MNIST-USPS* MEASURED BY ACCURACY AND NMI.

View	Percentage	Accuracy						
		MaxVar	LS	MCFS	UDFS	NDFS	MVFS	RMFS
View1	0.1	**0.4653 ± 0.0308**	0.4089 ± 0.0204	0.4593 ± 0.0343	0.4207 ± 0.0307	0.4140 ± 0.0212	0.1851 ± 0.0104	0.2001 ± 0.0035
	0.3	**0.5209 ± 0.0637**	0.5053 ± 0.0386	0.4941 ± 0.0315	0.4847 ± 0.0410	0.5140 ± 0.0458	0.3089 ± 0.0228	0.4109 ± 0.0250
	0.5	0.4953 ± 0.0506	0.4963 ± 0.0466	0.4867 ± 0.0274	**0.5213 ± 0.0239**	0.5046 ± 0.0496	0.4156 ± 0.0290	0.5148 ± 0.0390
	0.7	0.5161 ± 0.0380	0.4927 ± 0.0437	0.5179 ± 0.0499	0.5206 ± 0.0583	0.5024 ± 0.0559	0.4955 ± 0.0598	**0.5333 ± 0.0399**
	0.9	0.5006 ± 0.0415	0.5153 ± 0.0596	0.4826 ± 0.0476	0.5201 ± 0.0441	0.5081 ± 0.0347	0.5196 ± 0.0481	**0.5558 ± 0.0405**
View2	0.1	**0.6068 ± 0.0281**	0.5126 ± 0.033	0.6009 ± 0.0347	0.2788 ± 0.0059	0.4923 ± 0.0123	0.4361 ± 0.0271	0.4426 ± 0.0231
	0.3	0.6660 ± 0.0361	0.6124 ± 0.0275	**0.6696 ± 0.0425**	0.3837 ± 0.0142	0.5679 ± 0.0245	0.5911 ± 0.0283	0.5707 ± 0.0255
	0.5	0.6377 ± 0.0382	0.6666 ± 0.0160	0.6620 ± 0.0556	0.5163 ± 0.0295	0.6585 ± 0.0267	0.6351 ± 0.0253	**0.6694 ± 0.0113**
	0.7	0.6763 ± 0.0332	0.6726 ± 0.0378	0.6823 ± 0.0249	0.5620 ± 0.0188	0.6706 ± 0.0615	0.6578 ± 0.0431	**0.6889 ± 0.0480**
	0.9	0.6745 ± 0.0322	0.6664 ± 0.0295	0.6601 ± 0.0385	0.6103 ± 0.0408	0.6697 ± 0.0337	0.6657 ± 0.0329	**0.6768 ± 0.0333**
		NMI						
View1	0.1	**0.4256 ± 0.0158**	0.4003 ± 0.0124	0.4246 ± 0.0087	0.3373 ± 0.0091	0.3779 ± 0.0200	0.1225 ± 0.0161	0.1307 ± 0.0024
	0.3	**0.5196 ± 0.0252**	0.4883 ± 0.0171	0.5176 ± 0.0119	0.4750 ± 0.0177	0.5196 ± 0.0226	0.2488 ± 0.0194	0.3743 ± 0.0157
	0.5	0.5010 ± 0.0273	0.5044 ± 0.0256	0.4977 ± 0.0203	**0.5185 ± 0.0139**	0.5076 ± 0.0281	0.3741 ± 0.0238	0.5171 ± 0.0220
	0.7	0.5132 ± 0.0208	0.5069 ± 0.0190	0.5162 ± 0.0221	0.5217 ± 0.0254	0.5252 ± 0.0271	0.5133 ± 0.0254	**0.5273 ± 0.0200**
	0.9	0.5098 ± 0.0277	0.5164 ± 0.0289	0.5074 ± 0.0230	0.5123 ± 0.0214	0.5063 ± 0.0146	0.5158 ± 0.0249	**0.5270 ± 0.0170**
View2	0.1	**0.5226 ± 0.0144**	0.4556 ± 0.0138	0.5369 ± 0.0167	0.2354 ± 0.0055	0.4392 ± 0.0083	0.4462 ± 0.0310	0.3891 ± 0.0145
	0.3	**0.6094 ± 0.0185**	0.5811 ± 0.0144	0.5951 ± 0.0206	0.3083 ± 0.0069	0.5261 ± 0.0090	0.5591 ± 0.0256	0.5164 ± 0.0081
	0.5	**0.6227 ± 0.0119**	0.6197 ± 0.0123	0.6093 ± 0.0206	0.4655 ± 0.0166	0.6132 ± 0.0130	0.5922 ± 0.0147	0.5940 ± 0.0110
	0.7	0.6095 ± 0.0118	0.6084 ± 0.0204	0.6016 ± 0.0209	0.5202 ± 0.0112	**0.6134 ± 0.0254**	0.6087 ± 0.0193	0.6113 ± 0.0141
	0.9	0.6069 ± 0.0167	0.6084 ± 0.0128	0.6013 ± 0.0163	0.5703 ± 0.0218	0.6043 ± 0.0215	0.5948 ± 0.0259	**0.6108 ± 0.0139**

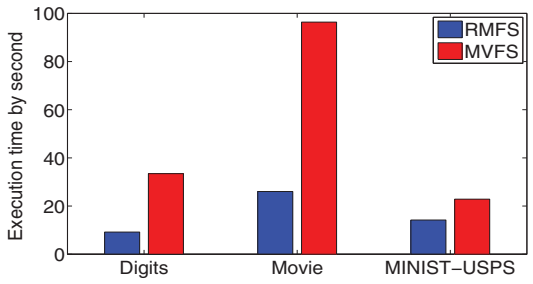


Figure 3. Execution time of RMFS and MVFS by second (10 runs).

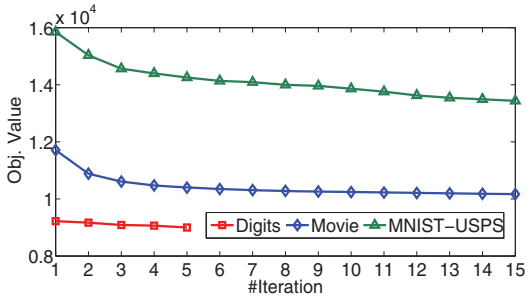


Figure 4. Convergence study of RMFS on three data sets.

from View-2) is much better than the the best result provided by MVFS ($NMI = 0.6154$, with 30% features from View-1 and 50% features from View-2) and RMFS outperforms MVFS over 50% in terms of NMI in the setting with 10% features from View-1 and 90% features from View-2. Recall that the performance of RMVC on single-view evaluation on *MNIST-USPS* with low selected feature ratios is worse than the one provided by single-view feature selection method. When combining the selected features from different views, the performance is boosted a lot. This indicates that our method takes the features from all views into account and the combination of the selected features is jointly to improve the clustering performance.

In terms of efficiency, Figure 3 show the 10 runs execution time of RMFS and MVFS on these three data sets using a PC with two Intel Core i7 3.4GHz CPUs and 32 GB RAM. Our proposed method runs faster than MVFS by a large margin. This results from that multi-view K-means is employed to obtain the pseudo labels instead of multi-kernel learning, which needs the eigenvector decomposition. It indicates that RMFS is suitable for multi-view feature selection on large-scale data sets.

Finally we experimentally study the convergence of RMFS to verify the correctness of Theorem 2 by Figure 4, which shows the convergence curves of three data sets. Generally speaking, RMFS converges fast within 15 iterations, which demonstrates high quality pseudo labels generated from multi-view learning are conductive to accelerate the convergence speed of RMFS. There is only one parameter β in our model to control the sparsity. The parameter β is not very sensitive within $[1e-3, 1e+3]$. Due to the limited pages, we omit the parameter analysis here.

VI. CONCLUSION

In this paper, we proposed a novel algorithm named Robust Muli-view Feature Selection (RMFS) for multi-view unsupervised feature selection, which provided robust and high quality pseudo labels from multi-view learning to guide the feature selection process and had much lower time complexity than existing methods. Further, a K-means-like optimization solution was designed on an augmented matrix to update several variables in a unified framework. Extensive experiments on three real-world data sets revealed that the effectiveness of RMFS in terms of both single-view and multi-view evaluations and the fast convergence speed.

VII. Acknowledgement

This work is supported in part by the NSF IIS Award 1651902, NSF CNS Award 1314484, ONR Award N00014-12-1-1028, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Young Investigator Award W911NF-14-1-0218.

References

[1] S. Bickel and T. Scheffer. Multi-view clustering. In *Proceedings of International Conference on Data Mining*, 2004.

[2] J. Bins and A. Draper. Feature selection from huge feature sets. In *Proceedings of International Conference on Computer Vision*, 2001.

[3] D. Cai, C. Zhang, and X. He. Unsupervised feature selection for multi-cluster data. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 2010.

[4] X. Cai, F. Nie, and H. Huang. Multi-view k-means clustering on big data. In *Proceedings of International Joint Conference on Artificial Intelligence*, 2013.

[5] N. Chen, J. Zhu, and E. Xing. Predictive subspace learning for multi-view data: a large margin approach. In *Proceedings of Advances in Neural Information Processing Systems*, 2010.

[6] C. Constantinopoulos, M. Titsias, and A. Likas. Bayesian feature and model selection for gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):1013–1018, 2006.

[7] Z. Ding and Y. Fu. Low-rank common subspace for multi-view learning. In *Proceedings of IEEE International Conference on Data Mining*, 2014.

[8] Z. Ding and Y. Fu. Robust multi-view subspace learning through dual low-rank decompositions. In *Proceedings of AAAI Conference on Artificial Intelligence*, 2016.

[9] D. Erhan, Y. Bengio, A. Courville, P. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *The Journal of Machine Learning Research*, 11:625–660, 2010.

[10] Y. Feng, J. Xiao, Y. Zhuang, and X. Liu. Adaptive unsupervised multi-view feature selection for visual concept recognition. In *Proceedings of Asian Conference on Computer Vision*, 2013.

[11] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[12] Y. Guo. Convex subspace representation learning from multi-view data. In *Proceedingse of AAAI Conference on Artificial Intelligence*, 2013.

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

[14] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *Proceedings of Advances in Neural Information Processing Systems*, 2005.

[15] S. Hong, Y. Li, Y. Han, and Q. Hu. Cluster structure preserving unsupervised feature selection for multi-view tasks. *Neurocomputing*, 17:686–697, 2016.

[16] A. Jovi, K. Brki, and N. Bogunovi. A review of feature selection methods with applications. In *Proceedings of International Convention on Information and Communication Technology, Electronics and Microelectronics*, 2015.

[17] Y. Kim, W. Street, and F. Menczer. Evolutionary model selection in unsupervised learning. *Intelligent data analysis*, 6(6):531–556, 2002.

[18] L. Kuncheva and W. Faithfull. Pca feature extraction for change detection in multidimensional unlabeled data. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1):69–80, 2014.

[19] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu. Unsupervised feature selection using nonnegative spectral analysis. In *Proceedings of AAAI Conference on Artificial Intelligence*, 2012.

[20] H. Liu, T. Liu, J. Wu, D. Tao, and Y. Fu. Spectral ensemble clustering. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2015.

[21] H. Liu, M. Shao, and Y. Fu. Consensus guided unsupervised feature selection. In *Proceedings of AAAI Conference on Artificial Intelligence*, 2016.

[22] H. Liu, M. Shao, S. Li, and Y. Fu. Infinite ensemble for image clustering. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2016.

[23] T. Liu, M. Gong, and D. Tao. Large cone nonnegative matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems*, 2016.

[24] T. Liu, D. Tao, and D. Xu. Dimensionality-dependent generalization bounds for k-dimensional coding schemes. *arXiv preprint arXiv:1601.00238*, 2016.

[25] S. Mahajan and S. Singh. Review on feature selection approaches using gene expression data. *Imperial Journal of Interdisciplinary Research*, 2(3), 2016.

[26] J. Mairal and B. Yu. Supervised feature selection in graphs with path coding penalties and network flows. *The Journal of Machine Learning Research*, 14(1):2449–2485, 2013.

[27] F. Nie, H. Huang, X. X. Cai, and C. Ding. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *Proceedings of Advances in Neural Information Processing Systems*, 2010.

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

[29] S. Sun. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7):2031–2038, 2013.

[30] J. Tang, X. Hu, H. Gao, and H. Liu. Unsupervised feature selection for multi-view data in social media. In *Proceedings of SIAM conference on Data Mining*, 2013.

[31] H. Wang, F. Nie, and H. Huang. Multi-view clustering and feature learning via structured sparsity. In *Proceedings of International Conference on Machine Learning*, 2013.

[32] J. Wu, H. Xiong, and J. Chen. Adapting the right measures for k-means clustering. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2009.

[33] W. Yang, Y. Gao, Y. Shi, and L. Cao. Mrm-lasso: A sparse multiview feature selection method via low-rank analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 26(9):2801–2815, 2015.

[34] Y. Yang, H. Shen, Z. Ma, Z. Huang, and X. Zhou. l2, 1-norm regularized discriminative feature selection for unsupervised learning. In *Proceedings of International Joint Conference on Artificial Intelligence*, 2011.

[35] C. Ying, X. Fern, and G. Jennifer. Non-redundant multi-view clustering via orthogonalization. In *Proceedings of International Conference on Data Mining*, 2007.

[36] D. Zhang, F. Wang, C. Zhang, and T. Li. Multi-view local learning. In *Proceedingse of AAAI Conference on Artificial Intelligence*, 2008.

[37] Z. Zhao, X. He, D. Cai, L. Zhang, W. Ng, and Y. Zhuang. Graph regularized feature selection with data reconstruction. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):689–700, 2016.

[38] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of International Conference on Machine Learning*, 2007.