MRM-Lasso: A Sparse Multi-View Feature Selection Method via Low-Rank Analysis

Wanqi Yang, Yang Gao*, *Member, IEEE,* Yinghuan Shi, Longbing Cao, *Senior Member, IEEE*

***Abstract*—Learning about multi-view data involves many applications, e.g., video understanding, image classification, and social media. However, when the data dimension increases dramatically, it is important but very challenging to remove redundant features in multi-view feature selection. In this paper, we propose a novel feature selection algorithm, Multi-view Rank Minimization-based Lasso (MRM-Lasso), which jointly utilizes Lasso for sparse feature selection and rank minimization for learning relevant patterns across views. Instead of simply integrating multiple Lasso from view-level, we focus on the performance of sample-level (sample significance) and introduce pattern-specific weights into MRM-Lasso. The weights are utilized to measure the contribution of each sample to the labels in the current view. Also, the latent correlation across different views is successfully captured by learning a low-rank matrix consisting of pattern-specific weights. The alternating direction method of multipliers (ADMM) is applied to optimize the proposed MRM-Lasso. Experiments on four real-life datasets show that features selected by MRM-Lasso have better multi-view classification performance than the baselines. Moreover, pattern-specific weights are demonstrated to be significant for learning about multi-view data, compared with view-specific weights.**

***Index Terms*—sparse multi-view feature selection, Lasso, low-rank matrix, pattern-specific weights**

I. Introduction

MULTI-VIEW learning captures the relationships across multiple views to improve the performance of classification or clustering [1]. As a promising machine learning tool, many multi-view learning algorithms [2], [3], [4], [5], [6], [7] have been successfully applied to handle relatively low-dimensional multi-view data. However, they are not effective for analyzing high-dimensional multi-view data due to the curse of dimensionality. In particular, lots of redundant features in high-dimensional data aggravate the difficulty of learning the intrinsic relationships across views.

There have been several methods proposed for solving the high-dimensional problem of multi-view learning, including multi-view dimensionality reduction [8], multi-view subspace learning [9], multi-view metric learning [10], etc. Most approaches simply focus on learning a shared subspace across views to connect all views but fail to select relevant and significant features from multiple different feature spaces (the selected features may represent the physical meanings). Thus, feature selection has been a popular research topic to find the relevant features. However, many feature selection algorithms [11], [12], [13], proposed for single-view data, are not applied to multi-view data, since both (1) the simple ensemble strategy of each-view feature selection and (2) integrating all views into a single one for single-view feature selection, ignore the relationships across views.

Accordingly, most recent efforts [14], [15], [16] have been made on sparse multi-view feature selection, which simultaneously selects the most discriminative features from multi-views through learning the relationships across views. Both Feng *et al.*'s method [14] and Tang *et al.*'s method [15] utilized multi-view shared clustering labels and view-specific weights to measure the relationship across views, and selected multi-view features via $L_{2,1}$-norm regularization. Wang *et al.* [16] selected different multi-view features for different clusters via joint group-sparsity and $L_{2,1}$-norm regularization. In all, the three methods [14], [15], [16] consider both view significance and feature significance, and the latter method [16] measures different feature significance in terms of every individual cluster. However, all of them might ignore sample significance, which represents the discrimination capability of samples to class labels.

'kick' in View 1 'kick' in View 4 'wave' in View 1 'wave' in View 4

Fig. 1. The typical images of actions 'kick' and 'wave' from two of the five views in IXMAS dataset[1]: (a) and (b) depict action 'kick' in View 1 and View 4 respectively; (c) and (d) depict action 'wave' in View 1 and View 4 respectively.

In many real situations, in fact, different samples in multi-view have varied discrimination capability. Sample significance analysis is very important for learning the detailed connection across views. One example is on multi-camera human activity recognition (see Fig. 1). View 4 can identify the action 'kick' better than View 1 because of the obvious shape features of leg extension in View 4, while View 1 has a stronger recognition capability for the action 'wave' because of its positive beckoning features. In other words, in View 4, 'kick' is more discriminative than 'wave', while 'kick' performs worse than 'wave' in View 1. To select the shape features of leg extension (in View 4) and the beckoning features (in View 1), the discriminative samples which can best describe these features in the current view should be assigned as the topmost weight values for measuring the importance of these features. On the contrary, those insignificant or noisy samples in the current view should be assigned negligible

Manuscript received xx xx, 201x; revised xx xx, 201x. This work was supported in part by the National Science Foundation of China under (Grant Nos. 61432008, 61305068, 61321491), the Graduate Research Innovation Program of Jiangsu, China (CXZZ13_0055), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

Wanqi Yang, Yang Gao (Corresponding Author) and Yinghuan Shi are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210046, China. (e-mail: nju.yangwanqi@gmail.com, gaoy@nju.edu.cn and syh@nju.edu.cn)

Longbing Cao is with the Advanced Analytics Institute, University of Technology Sydney, Australia. (e-mail: longbing.cao@uts.edu.au)

[1] http://4drepository.inrialpes.fr/public/viewgroup/6

weight values, since they will make it difficult to find the most discriminative features. Here, a sample in a particular view is namely a pattern. The weights are namely pattern-specific weights (or pattern weights).

Therefore, instead of treating different samples equally (using the same weight for a certain view), a promising alternative is to assume that different samples usually have different weights according to their sample significance. Specifically, we measure the sample significance by introducing pattern-specific weights, which can alleviate the measurement error for feature significance caused by less discriminative or noisy patterns in the current view.

Basically, all pattern-specific weights can be grouped into a pattern weight matrix, where each column represents the pattern weights of a certain view. (1) Sparsity: the pattern weight matrix is expected to be sparse in order to weaken the role of less discriminative patterns and enhance the importance of more discriminative patterns; (2) Cross-view correlation: the pattern weight matrix is expected to be relevant across different columns, since similar patterns have similar pattern-specific weights, and the cross-view correlation can be addressed by analyzing the corresponding relevant patterns across views.

Thus, we impose the low-rank constraint on the pattern weight matrix, which not only reflects the sparsity of pattern weights (sample significance), but also captures the related patterns across views (view significance).

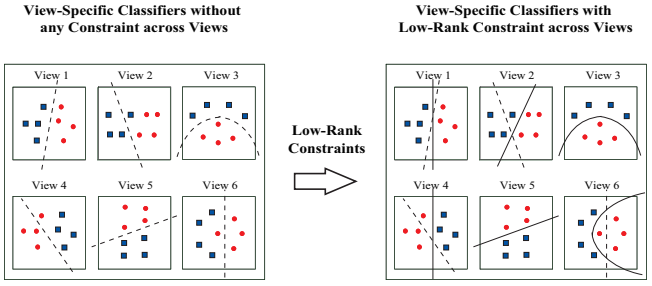


Fig. 2. An example of six-view data with a low-rank weight matrix, where the squares and circles represent samples in different classes, respectively. The left part presents six view-specific classifiers learned individually without any constraint across views; the right part presents the updated results (solid lines) with the low-rank weight matrix constraint.

Fig. 2 illustrates the effect of the low-rank weight matrix on multi-view data. We can see how the relevant patterns across the correlated views help each other for better classification. From the data distribution (refer to the figure on the left), there are three pairs of correlated views: views 1 and 4, 2 and 5, and 3 and 6, respectively. Low-rank constraints on each pair of correlated views can capture the latent correlation; as a result, several misclassification problems can be well tackled (refer to the figure on the right). A low-rank matrix with its cross-row/column linear-correlation characteristics has been demonstrated to be effective in many applications, e.g., multi-task learning [17] and domain adaptation [18]. We first introduce the low rank in the multi-view feature selection for (1) measuring sample significance with respect to different views, and (2) capturing the related patterns across views.

For high-dimensional single-view data, sparse feature selection methods such as Lasso [19] and its variants [20], [21], [22], [23], [24] have been successfully applied to many applications, e.g., medical image analysis [25], face recognition [26] and social network analysis [27]. In this paper, Lasso is extended to a multi-view learning scenario by considering the correlation across views, where the most significant features of multi-views can be simultaneously selected.

Therefore, we propose Multi-view Rank Minimization-based Lasso (MRM-Lasso) for sparse multi-view feature selection, which jointly integrates sparse learning and the low-rank pattern-specific weight matrix. The proposed method considers three aspects: view significance, feature significance and sample significance. Our main contributions are three-fold, as follows:

- The pattern-specific weights are first adopted to measure the discrimination of different patterns (sample significance). Consequently, a low-rank constraint encodes the pattern weight matrix to capture the relevant patterns across views.
- Lasso is first extended for sparse multi-view feature selection (feature significance) and relevant pattern analysis (view significance).
- A novel multi-view feature selection method, MRM-Lasso, is proposed. Also, a feasible alternating optimization strategy is applied via the Alternating Direction Method of Multipliers (ADMM).

In Section II, the related work is discussed. The background of sparse feature selection is described in Section III. Section IV introduces our proposed MRM-Lasso. The experimental results on the classification performance of MRM-Lasso pattern-specific weights are presented in Section V and VI, respectively, followed by the conclusions in Section VII.

II. Related Work

A. Feature Selection in the Single View

Recently, many feature selection algorithms [11], [12], [13] have been proposed, which can be generally classified into three kinds: wrapper, filter and embedded methods [28]. The wrapper methods [29] evaluate the importance of features by estimating the performance of learned classifiers; the filter methods [30] select features according to a particular correlation criteria; the embedded methods [31] have a one-step learning process by combining feature selection and classification. Of the three kinds of algorithms, we focus on the filter methods, which are the least time-consuming as they do not require classifier training during the feature selection process. For the filter methods, the optimal feature subset is selected via the importance of the feature subset, which is evaluated by the relevance of features to class labels, e.g., ReliefF [32], Fisher score [33] and mRMR [34].

However, a simple extension (simple ensemble strategy across views or integrating all views into a single one) of traditional single-view feature selection algorithms cannot be directly utilized for multi-view feature selection, since they fail to consider the cross-view connection. According to our knowledge, research on multi-view feature selection has rarely been conducted [14], [15], [16].

B. The Connection across Views

In multi-view learning, the connection across views has been considered to improve the performance of learning tasks, e.g., multi-view classification, multi-view clustering/subspace learning, and multi-view feature selection. The existing methods can be divided into three categories as follows.

TABLE I
NOTATIONS USED IN THIS PAPER

Notations	Descriptions
$\mathcal{X}_i; y_i \in \mathbb{R}, \mathbf{x}_i^v \in \mathbb{R}^{n_v}$	multi-view training sample, the corresponding class label and every view-specific pattern
$\mathbf{y}, \mathbf{y}_k \in \mathbb{R}^m$	the vector of training class labels and that of the k^{th} task in multi-task Lasso
$\mathcal{Z}; \mathbf{z}^v \in \mathbb{R}^{n_v}$	multi-view testing sample preprocessed by feature selection and every view-specific pattern
$\mathbf{F} \in \mathbb{R}^{m \times n}, \mathbf{F}_k \in \mathbb{R}^{m \times n^k}$	feature matrix of single-view data and that of the k^{th} group (or task) in group Lasso (or multi-task Lasso)
$\mathbf{F}^v \in \mathbb{R}^{m \times n_v}$	view-specific feature matrix of the v^{th} view
$\beta_j \in \mathbb{R}, \boldsymbol{\beta} \in \mathbb{R}^n$	feature selection coefficient for the j^{th} feature in single-view data and its feature selection coefficient vector
$\beta_{kj} \in \mathbb{R}, \boldsymbol{\beta}_k \in \mathbb{R}^n$	feature selection coefficient for the j^{th} feature in the k^{th} group (or task) and its feature selection coefficient vector
$\boldsymbol{\beta}^v \in \mathbb{R}^{n_v}$	view-specific feature selection coefficient vector of the v^{th} view
$\alpha^v, w_i^v \in \mathbb{R}$	view-specific weight and pattern-specific weight in the v^{th} view
$\mathbf{W} \in \mathbb{R}^{m \times s}, \mathbf{w}^v \in \mathbb{R}^m$	weight matrix constituted by w_i^v and its v^{th} column vector
$\mathbf{J}, \mathbf{\Lambda} \in \mathbb{R}^{m \times s}, \mathbf{u}_i^v \in \mathbb{R}^{n_v}$	new variables introduced for the solution of $\mathbf{W}$ and $\boldsymbol{\beta}^v$ in Eq.(6)
$\mathbf{U}^v \in \mathbb{R}^{m \times n_v}, \mathbf{U} \in \mathbb{R}^{m \times N}$	view-specific matrix constituted by $\{\mathbf{u}_i^v\}_{i=1}^m$ and the cross-view form spliced by $\{\mathbf{U}^v\}$
$\mathbf{P}^v \in \mathbb{R}^{m \times n_v}, \mathbf{Q}, \mathbf{R} \in \mathbb{R}^{m \times s}$	the Lagrange multipliers in the augmented Lagrangian form
$\mathbf{U}_i, \mathbf{P}_i, \mathbf{C}_i \in \mathbb{R}^N$	three vectors spliced by the i^{th} row vectors of $\{\mathbf{P}^v\}$ and $\{\mathbf{U}^v\}$, and $\{w_i^v \boldsymbol{\beta}^v\}_{v=1}^s$, respectively
$\rho, \mu, \xi \in \mathbb{R}$	three parameters of augmented Lagrange terms
$\lambda_S, \lambda_{S'}, \lambda_F, \lambda_R \in \mathbb{R}$	the parameters of the sparsity term, intragroup-sparsity term, smoothness term, and low-rank term
$m, s, n, n_v, N, K \in \mathbb{R}$	the number of samples, views, single-view features, the v^{th}-view features, all view features and tasks (or groups)

The first class of methods is to impose a consistent constraint on the prediction labels of view-specific classifiers. Based on such an assumption [2] (multiple views are conditionally independent given the class label), most multi-view classification algorithms [3], [4], [5] learn a series of view-specific classifiers together with the constraint of consistent prediction labels, and then ensemble their results for classification.

The second class of methods is to exploit the latent variables or subspace shared by all views. Many algorithms [1], [6], [7] learn the shared subspace via several different methods, e.g., canonical correlation analysis [1], tensor methods [6] and conditional random fields [7].

The third class of methods is to learn a particular metric across different views. Many algorithms [10], [35], [36] aim to seek a shared latent feature space that can shorten the distance of the mapped data between different views, and then learn the mapping functions between the input spaces and the shared latent space.

However, the above three categories learn the connection across views from view-level (view significance), but fail to consider different discriminations and relevances from sample-level (sample significance).

C. Application of Low-Rank Matrix

The methodology of a low-rank matrix or matrix rank minimization has aroused the interest of researchers recently. Matrix recovery and correlation analysis of a low-rank matrix have been widely utilized in many applications, e.g., face recognition [37], [38], image recovery [39], subspace segmentation [40], [41], multi-task learning [17], and domain adaptation [18], etc. For low-rank matrix recovery, Wright *et al.* [39] proposed Robust PCA to recover the matrix, where the low-rank term and the sparsity term are jointly minimized. For robust subspace segmentation, Liu *et al.* [40] imposed low-rank structure on the linear representation matrix of data. Luo *et al.* [41] learned multi-subspace representation via solving a low-rank and sparse matrix, when data points belong to multiple independent subspaces. For face recognition, Ma *et al.* [37] proposed a discriminative low-rank dictionary learning algorithm for sparse representation, where the learned sparse coefficients are utilized for face recognition. For the correlation analysis of a low-rank matrix, Ye *et al.* [17] fused the predicted confidence scores of multiple models via seeking a shared rank-2 pairwise relationship matrix for multi-task learning. In domain adaptation, in order to find an intermediate representation correlated with the source domain and target domain, Jhuo *et al.* [18] presented a low-rank reconstruction method to reduce the domain distribution disparity.

However, as these methods are only designed for the single-view learning task, our proposed method is the first attempt to introduce the low rank in the multi-view feature selection for (1) measuring sample significance with respect to different views, and (2) capturing the related patterns across views.

D. Connection to Multiple Kernel Learning

According to ensemble learning theories [42], Multiple Kernel Learning (MKL) forms an ensemble of multiple kernels for better generalizability compared to a single fixed kernel [43]. Since MKL includes multiple kernels, it can be naturally applied to multi-view data fusion. MKL aims to improve classification performance, which measures the significance of different views by combining multiple kernels from multi-views. However, our method is utilized for multi-view feature selection, which not only measures view significance, but also considers both feature significance and sample significance. They are very useful for interpreting the domain-specific meanings of different multi-view data.

E. Connection to Existing Multi-View Feature Selection Methods

Existing multi-view feature selection methods [14], [15], [16] are 'global' methods, while our method is a 'global and local' method. Specifically, the work in [14], [15] utilized view-specific weights and feature selection coefficients to measure view significance and feature significance, respectively. The work in [16] measured different feature significance and view significance in terms of every individual cluster. Since the above three methods ignore sample significance, they belong to the 'global' method. However, our method not only focuses on feature significance and view significance, but also

emphasizes sample significance (not mentioned in [14], [15], [16]), which covers both 'global and local' aspects. Sample significance can help us find the most discriminative samples to enhance feature selection performance, since treating all the different samples equally is not a good choice (ignored in [14], [15], [16]), which has been validated in our experiments in Section VI.

III. Background

A. Notations

Below, a bold upper case letter denotes a matrix, a bold lower case letter denotes a vector, an upper case decorated letter represents a set, and a normal lower case letter corresponds to a scalar.

In this paper, we assume that every multi-view sample has a common class label across views[2]. There are m labeled samples $\{\mathcal{X}_i, y_i\}_{i=1}^m$, all of which have s view patterns, i.e., $\mathcal{X}_i = \{\mathbf{x}_i^1, \mathbf{x}_i^2, ..., \mathbf{x}_i^s\}$ ($\mathbf{x}_i^v \in \mathbb{R}^{n_v}$). Their subscripts indicate the index of instances, while the superscripts are the view indices. For each view, the feature dimension is denoted as n_v ($N = \sum_{v=1}^s n_v$). The view-specific feature matrix is denoted as $\mathbf{F}^v \in \mathbb{R}^{m \times n_v}$, where $\mathbf{F}^v = (\mathbf{x}_1^v, \mathbf{x}_2^v, ..., \mathbf{x}_m^v)^\top$. Table I records the main notations used in this paper.

B. Sparse Feature Selection

One of the most popular sparse feature selection methods, Lasso, was originally proposed for feature selection by Robert Tibshirani in [19]. For single-view data, $\mathbf{F} \in \mathbb{R}^{m \times n}$ and $\mathbf{y} \in \mathbb{R}^m$ represent a feature matrix and the corresponding label vector. $\boldsymbol{\beta} \in \mathbb{R}^n$ represents a vector of feature selection coefficients, which measures the importance of the corresponding features. The objective problem can be written as

$$\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{F}\boldsymbol{\beta}||_2^2 + \lambda_S ||\boldsymbol{\beta}||_1, \tag{1}$$

where $\lambda_S \in \mathbb{R}$ is the sparsity parameter. The above equation minimizes the L_1-regularized reconstruction error to solve $\boldsymbol{\beta}$. Then the features with non-zero values of $\boldsymbol{\beta}$ will be selected.

Afterwards, many variants of Lasso were explored, e.g., group Lasso [20], [21] and fused Lasso [22]. Group Lasso methods are designed for data with group structures, where all features are partitioned into K groups. $\mathbf{F}_k \in \mathbb{R}^{m \times n^k}$ and $\boldsymbol{\beta}_k \in \mathbb{R}^{n^k}$ are the feature matrix of the k^{th} group and the corresponding feature selection coefficient vector, respectively. Group Lasso [20] utilizes an L_2-regularized term for each group to yield intergroup sparsity, whose formulation is written as follows:

$$\min_{\boldsymbol{\beta}_k} ||\mathbf{y} - \sum_{k=1}^K \mathbf{F}_k\boldsymbol{\beta}_k||_2^2 + \lambda_S \sum_{k=1}^K ||\boldsymbol{\beta}_k||_2. \tag{2}$$

Adding to the intragroup sparsity, the sparse group Lasso [21] is updated as

$$\min_{\boldsymbol{\beta}_k} ||\mathbf{y} - \sum_{k=1}^K \mathbf{F}_k\boldsymbol{\beta}_k||_2^2 + \lambda_S \sum_{k=1}^K ||\boldsymbol{\beta}_k||_2 + \lambda_{S'} ||\boldsymbol{\Omega}||_1, \tag{3}$$

[2]We do not consider the cases of label non-consistency across views or missing labels.

where $\boldsymbol{\Omega} = (\boldsymbol{\beta}_1^\top, \boldsymbol{\beta}_2^\top, ..., \boldsymbol{\beta}_K^\top)^\top$, and $\lambda_{S'} \in \mathbb{R}$ is the parameter of the intragroup sparsity term.

Different from group Lasso, fused Lasso [22] is applied to the data with a temporal or spacial relationship. The objective function of fused Lasso minimizes the difference of adjacent β_j and β_{j-1} by imposing the neighboring smoothness constraint.

$$\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{F}\boldsymbol{\beta}||_2^2 + \lambda_S ||\boldsymbol{\beta}||_1 + \lambda_F \sum_{j=2}^n |\beta_j - \beta_{j-1}|, \tag{4}$$

where the third term is the smoothness term and λ_F is the smoothness parameter.

In multi-task feature selection, all tasks are encouraged to contain the common sparsity [23], [24]. There are K tasks, and $\mathbf{F}_k \in \mathbb{R}^{m \times n^k}$ is the feature matrix of the k^{th} task. In most cases, the data dimension of different tasks is assumed to be equivalent, where $n^k = n,\ k = 1, .., K$. Let $\beta_{kj} \in \mathbb{R}$ be the feature selection coefficient of the j^{th} feature in the k^{th} task, then the feature selection coefficient vector $\boldsymbol{\beta}_k = (\beta_{k1}, \beta_{k2}, ..., \beta_{kn})^\top \in \mathbb{R}^n$. Multi-task Lasso [23] is formulated as

$$\min_{\boldsymbol{\beta}_k} \sum_{k=1}^K ||\mathbf{y}_k - \mathbf{F}_k\boldsymbol{\beta}_k||_2^2 + \lambda_S \sum_{j=1}^n ||\boldsymbol{\eta}_j||_2, \tag{5}$$

where $\boldsymbol{\eta}_j = (\beta_{1j}, \beta_{2j}, ..., \beta_{Kj})^\top \in \mathbb{R}^K$, and the second item shows the common sparsity of selected features across all tasks. In addition to common sparsity, task-specific sparsity is also considered for multi-task feature selection [24].

TABLE II
Sparse Feature Selection Methods

Methods	Data Types or Problems
Lasso	High dimension
Group Lasso	Group structure with intergroup sparsity
Sparse Group Lasso	Group structure with intergroup and intragroup sparsity
Fused Lasso	Locality or neighboring smoothness
Multi-task Lasso	Common sparsity across tasks
Multi-level Lasso	Common sparsity across tasks and task-specific sparsity

In summary, the above Lasso-based methods are proposed to deal with various data types or problems which are listed in Table II. When data has multiple views, however, it is challenging to utilize Lasso for multi-view feature selection. For each view, the vector of feature selection coefficients is denoted as $\boldsymbol{\beta}^v$. Note that selected features of $\boldsymbol{\beta}^v$ in the current view are completely different from features in the other views. Any direct constraints on $\{\boldsymbol{\beta}^v\}_{v=1}^s$, like Eqs.(2)-(5), are undesirable for multi-view data. In the following section, we will propose to extend Lasso for sparse multi-view feature selection and utilize a low-rank constraint to explore the connection across views.

IV. Problem Formulation: MRM-Lasso

We define the problem for sparse multi-view feature selection. When Lasso is directly utilized for sparse multi-view feature selection, features of multi-views can be easily combined to linearly represent the class label. The fitting term of the Lasso model in Eq.(1) is written as: $y_i = \sum_{v=1}^s \mathbf{x}_i^{v\top} \boldsymbol{\beta}^v + \epsilon$, where $\epsilon \in \mathbb{R}$ is the error term. However, the equation fails to measure the difference in individual patterns within and across views, which may result in a measurement error in the feature

importance due to less discriminative or noisy patterns in a certain view. To avoid this problem, we introduce pattern-specific weights $w_i^v \in \mathbb{R}$ into the feature selection model, whose fitting term can be rewritten as: $y_i = \sum_{v=1}^{s} w_i^v \cdot (\mathbf{x}_i^{v\top}\boldsymbol{\beta}^v) + \epsilon$, where $\sum_{v=1}^{s} w_i^v = 1,\ w_i^v > 0$. Weight w_i^v indicates the contribution of pattern $\mathbf{x}_i^v$ to the labels. Higher weight values will enhance the effect of discriminative patterns on measuring the feature importance. On the contrary, the patterns with lower weight values will have less impact on measuring the feature importance.

All of the weights $\{w_i^v\}$ can be collected to construct weight matrix $\mathbf{W} \in \mathbb{R}^{m\times s}$, where w_i^v is the $(i,v)^{\text{th}}$ element of matrix $\mathbf{W}$, and its column vector is denoted as $\mathbf{w}^v = (w_1^v, w_2^v, ..., w_m^v)^\top \in \mathbb{R}^m$. In our feature selection model, $\mathbf{W}$ and $\{\boldsymbol{\beta}^v\}$ are respectively utilized to investigate different patterns of multi-views and measure the importance of features from multiple feature spaces. Then, in the following sections, we will formulate the problem of multi-view feature selection and present the optimization strategy of learning $\mathbf{W}$ and $\{\boldsymbol{\beta}^v\}$.

A. MRM-Lasso

Based on the above considerations, we propose a novel multi-view feature selection method, MRM-Lasso, which can select features simultaneously from multi-views. The problem is formulated as follows:

$$\begin{aligned}\min_{\boldsymbol{\beta}^v,\mathbf{W}} \frac{1}{2m}\sum_{i=1}^{m}(y_i - \sum_{v=1}^{s} w_i^v \mathbf{x}_i^{v\top}\boldsymbol{\beta}^v)^2 + \lambda_S \sum_{v=1}^{s}||\boldsymbol{\beta}^v||_1 \\ + \lambda_R R(\mathbf{W}), \\ s.t.\ \forall\ i,\ \sum_{v=1}^{s} w_i^v = 1,\ i = 1,2,...,m \\ \forall\ i,v,\ w_i^v \geq 0,\ i = 1,2,...,m,\ v = 1,2,...,s\end{aligned} \tag{6}$$

where $\lambda_R \in \mathbb{R}$ and $\lambda_S \in \mathbb{R}$ are the regularization parameters. $R(\mathbf{W})$, the rank of $\mathbf{W}$, is minimized to measure the shared characteristics across views. The three terms in Eq.(6) affect the solution of variables $\mathbf{W}$ and $\boldsymbol{\beta}^v$ in different levels. For $\mathbf{W}$, the first term is the weighted reconstruction error between the ground-truth labels and their predictions across different views, where pattern weights of discriminative patterns are encouraged to be large in order to obtain small reconstruction error; while the third one encourages pattern weights to be more proportional across different patterns. For $\boldsymbol{\beta}^v$, the first term assigns a large value to discriminative features while the second one makes $\boldsymbol{\beta}^v$ sparse. The designed formulation aims to measure both view significance and sample significance by $\mathbf{W}$, and feature significance by $\{\boldsymbol{\beta}^v\}$.

B. Optimization Procedures for MRM-Lasso

Since the computation of the third term $R(\mathbf{W})$ in Eq.(6) is NP-hard, we approximately solve the nuclear norm $||\mathbf{W}||_*$, which is the sum of the singular values of matrix $\mathbf{W}$. The problem in Eq.(6) is non-convex, so we cannot find the optimal solution. A fast-convergent procedure is presented to repeatedly search the local optimum solutions of the MRM-Lasso problem to approximate the optimum one. We adopt the ADMM, which is more efficient than simple Alternative Convex Search (ACS). To change the problem in Eq.(6) into the ADMM form, we introduce variables $\mathbf{u}_i^v \in \mathbb{R}^{n_v}$, $\mathbf{J} \in \mathbb{R}^{m\times s}$ and nonnegative matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times s}$. Since the inequality-constraint problem using ADMM optimization can be converted into the equality-constraint problem by introducing several nonnegative slack variables [44], $\boldsymbol{\Lambda}$ is introduced to change the inequality constraint $\mathbf{W} \geq 0$ in Eq.(6) into the equality constraint, $-\mathbf{W} + \boldsymbol{\Lambda} = 0$. The objective can be reformulated as:

$$\begin{aligned}\min_{\boldsymbol{\beta}^v,\mathbf{W}} \frac{1}{2m}\sum_{i=1}^{m}(y_i - \sum_{v=1}^{s}\mathbf{x}_i^{v\top}\mathbf{u}_i^v)^2 + \lambda_S\sum_{v=1}^{s}||\boldsymbol{\beta}^v||_1 + \lambda_R||\mathbf{J}||_* \\ s.t.\quad \mathbf{W}\mathbf{1}_s = \mathbf{1}_m, -\mathbf{W} + \boldsymbol{\Lambda} = 0,\ \mathbf{J} = \mathbf{W}, \\ \mathbf{u}_i^v = w_i^v\boldsymbol{\beta}^v,\ v = 1,..,s,\ i = 1,...,m,\end{aligned} \tag{7}$$

where for each i, the vectors $\mathbf{u}_i^v \in \mathbb{R}^{n_v}$ can be spliced into a long vector $\mathbf{U}_i = (\mathbf{u}_i^{1\top}, ..., \mathbf{u}_i^{s\top})^\top \in \mathbb{R}^N$. All the $\mathbf{U}_i$ can be arranged into a matrix $\mathbf{U} = (\mathbf{U}_1, ..., \mathbf{U}_m)^\top \in \mathbb{R}^{m\times n}$. For each v, the $\mathbf{u}_i^v$ can be arranged into a matrix $\mathbf{U}^v = (\mathbf{u}_1^v, ..., \mathbf{u}_m^v)^\top \in \mathbb{R}^{m\times n_v}$, then $\mathbf{U}^v = \mathbf{w}^v\boldsymbol{\beta}^{v\top}$.

For notational simplicity, we denote the three terms in Eq.(7) as $f_1(\mathbf{U})$, $f_2(\{\boldsymbol{\beta}^v\})$ and $f_3(\mathbf{J})$, respectively. The augmented Lagrangian form can be written as

$$\begin{aligned}&L(\mathbf{W},\{\boldsymbol{\beta}^v\},\mathbf{U},\mathbf{J},\boldsymbol{\Lambda},\{\mathbf{P}^v\},\mathbf{Q},\mathbf{R}) \\ =&f_1(\mathbf{U}) + f_2(\{\boldsymbol{\beta}^v\}) + f_3(\mathbf{J}) + \langle\mathbf{Q},\boldsymbol{\Lambda}-\mathbf{W}\rangle + \frac{\mu}{2}||\boldsymbol{\Lambda}-\mathbf{W}||_F^2 \\ &+\sum_v\left\{\langle\mathbf{P}^v,\mathbf{w}^v\boldsymbol{\beta}^{v\top}-\mathbf{U}^v\rangle + \frac{\rho}{2}||\mathbf{w}^v\boldsymbol{\beta}^{v\top}-\mathbf{U}^v||_F^2\right\} \\ &+\langle\mathbf{R},\mathbf{J}-\mathbf{W}\rangle + \frac{\xi}{2}||\mathbf{J}-\mathbf{W}||_F^2\end{aligned}$$

where $\mathbf{P}^v \in \mathbb{R}^{m\times n_v}, \mathbf{Q} \in \mathbb{R}^{m\times s}, \mathbf{R} \in \mathbb{R}^{m\times s}$ are Lagrange multipliers and $\langle X_1, X_2\rangle$ means the trace of $X_1^\top X_2$,

Since the solution of $\boldsymbol{\Lambda}$ has the closed form $\boldsymbol{\Lambda}_{iv}^* = \max(0, w_i^v - (1/\mu)\mathbf{Q}_{iv})$, which can be introduced into the augmented Lagrangian form:

$$\begin{aligned}&L(\mathbf{W},\{\boldsymbol{\beta}^v\},\mathbf{U},\mathbf{J},\{\mathbf{P}^v\},\mathbf{Q},\mathbf{R}) \\ =&f_1(\mathbf{U}) + f_2(\{\boldsymbol{\beta}^v\}) + f_3(\mathbf{J}) + \frac{1}{2\mu}\sum_{iv}\Big\{\max(\mathbf{Q}_{iv}-\mu w_i^v, 0)^2 \\ &-\mathbf{Q}_{iv}^2\Big\} + \sum_v\left\{\langle\mathbf{P}^v,\mathbf{w}^v\boldsymbol{\beta}^{v\top}-\mathbf{U}^v\rangle + \frac{\rho}{2}||\mathbf{w}^v\boldsymbol{\beta}^{v\top}-\mathbf{U}^v||_F^2\right\} \\ &+\langle\mathbf{R},\mathbf{J}-\mathbf{W}\rangle + \frac{\xi}{2}||\mathbf{J}-\mathbf{W}||_F^2\end{aligned} \tag{8}$$

A general ADMM scheme is

$$\begin{cases}\mathbf{W} & \leftarrow \arg\min_{\mathbf{W}} L(\mathbf{W},\{\boldsymbol{\beta}^v\},\mathbf{U},\mathbf{J},\{\mathbf{P}^v\},\mathbf{Q},\mathbf{R}) \\ \{\boldsymbol{\beta}^v\} & \leftarrow \arg\min_{\{\boldsymbol{\beta}^v\}} L(\mathbf{W},\{\boldsymbol{\beta}^v\},\mathbf{U},\mathbf{J},\{\mathbf{P}^v\},\mathbf{Q},\mathbf{R}) \\ \mathbf{U} & \leftarrow \arg\min_{\mathbf{U}} L(\mathbf{W},\{\boldsymbol{\beta}^v\},\mathbf{U},\mathbf{J},\{\mathbf{P}^v\},\mathbf{Q},\mathbf{R}) \\ \mathbf{J} & \leftarrow \arg\min_{\mathbf{J}} L(\mathbf{W},\{\boldsymbol{\beta}^v\},\mathbf{U},\mathbf{J},\{\mathbf{P}^v\},\mathbf{Q},\mathbf{R}) \\ \mathbf{P}^v & \leftarrow \mathbf{P}^v + \rho(\mathbf{w}^v\boldsymbol{\beta}^{v\top} - \mathbf{U}^v) \\ \mathbf{Q}_{iv} & \leftarrow \max(\mathbf{Q}_{iv} - \mu w_i^v, 0) \\ \mathbf{R} & \leftarrow \mathbf{R} + \xi(\mathbf{J} - \mathbf{W})\end{cases} \tag{9}$$

where $v = 1, ..., s,\ i = 1, ..., m$. Note that every step in the ADMM scheme is a convex problem, so their optimal solutions in the current step can be definitely achieved. For notational simplicity, we denote $\mathbf{w}^v\boldsymbol{\beta}^{v\top} - \mathbf{U}^v$ and $w_i^v\boldsymbol{\beta}^v - \mathbf{u}_i^v$

as $\mathbf{\Gamma}^v \in \mathbb{R}^{m\times n_v}$ and $\boldsymbol{\gamma}_i^v \in \mathbb{R}^{n_v}$, respectively. The solutions of $\mathbf{W}$, $\{\boldsymbol{\beta}^v\}$, $\mathbf{U}$ and $\mathbf{J}$ are presented as follows.

1) **the solution of W**
The solution of $\mathbf{W}$ is

$$\begin{aligned}
&\arg\min_{\mathbf{W}} L(\mathbf{W},\{\boldsymbol{\beta}^v\},\mathbf{U},\mathbf{J},\{\mathbf{P}^v\},\mathbf{Q},\mathbf{R})\\
&=\arg\min_{\mathbf{W}} \frac{1}{2\mu}\sum_{iv}\Big[(\max(\mathbf{Q}_{iv}-\mu w_i^v,0))^2-\mathbf{Q}_{iv}^2\Big]\\
&\quad+\sum_v\Big[\langle\mathbf{P}^v,\mathbf{\Gamma}^v\rangle+\frac{\rho}{2}||\mathbf{\Gamma}^v||_F^2\Big]\\
&\quad+\langle\mathbf{R},\mathbf{J}-\mathbf{W}\rangle+\frac{\xi}{2}||\mathbf{J}-\mathbf{W}||_F^2\\
&=\arg\min_{\mathbf{W}}\sum_v\Big\{\frac{1}{2\mu}\sum_i\Big[(\max(\mathbf{Q}_{iv}-\mu w_i^v,0))^2-\mathbf{Q}_{iv}^2\Big]\\
&\quad+\langle\mathbf{P}^v,\mathbf{\Gamma}^v\rangle+\frac{\rho}{2}||\mathbf{\Gamma}^v||_F^2+\mathbf{R}_{\cdot v}^\top(\mathbf{J}_{\cdot v}-\mathbf{w}^v)\\
&\quad+\frac{\xi}{2}||\mathbf{J}_{\cdot v}-\mathbf{w}^v||^2\Big\}
\end{aligned}$$

where $\mathbf{R}_{\cdot v}$ and $\mathbf{J}_{\cdot v}$ are the column vectors of $\mathbf{R}$ and $\mathbf{W}$, respectively. It can be easily decomposed into multiple subproblems with respect to $\mathbf{w}^v$:

$$\begin{aligned}
&\arg\min_{\mathbf{w}^v}\frac{1}{2\mu}\sum_i\Big[(\max(\mathbf{Q}_{iv}-\mu w_i^v,0))^2-\mathbf{Q}_{iv}^2\Big]\\
&\quad+\langle\mathbf{P}^v,\mathbf{\Gamma}^v\rangle+\frac{\rho}{2}||\mathbf{\Gamma}^v||_F^2+\mathbf{R}_{\cdot v}^\top(\mathbf{J}_{\cdot v}-\mathbf{w}^v)\\
&\quad+\frac{\xi}{2}||\mathbf{J}_{\cdot v}-\mathbf{w}^v||^2
\end{aligned}\tag{10}$$

The above equation is a convex problem with respect to $\mathbf{w}^v$, which can be easily solved via gradient descent. Then the solution of $\mathbf{W}$ is normalized along rows.

2) **the solution of $\{\boldsymbol{\beta}^v\}$**
For each view, $\boldsymbol{\beta}^v$ can be solved by:

$$\begin{aligned}
&\arg\min_{\boldsymbol{\beta}^v} f_2(\boldsymbol{\beta}^v)+\langle\mathbf{P}^v,\mathbf{\Gamma}^v\rangle+\frac{\rho}{2}||\mathbf{\Gamma}^v||_F^2\\
&=\arg\min_{\boldsymbol{\beta}^v}\lambda_S||\boldsymbol{\beta}^v||_1+\langle\mathbf{P}^v,\mathbf{\Gamma}^v\rangle+\frac{\rho}{2}||\mathbf{\Gamma}^v||_F^2\\
&=\mathcal{S}_\epsilon[(\rho\mathbf{U}^v-\mathbf{P}^v)^\top\mathbf{w}^v/(d\rho)]
\end{aligned}\tag{11}$$

where $d=||\mathbf{w}^v||^2$, $\epsilon=\lambda_S/(d\rho)$ and $\mathcal{S}_\epsilon$ is the soft-thresholding (shrinkage) operator [45].

3) **the solution of U**
The solution of $\mathbf{U}$ is

$$\begin{aligned}
&\arg\min_{\mathbf{U}} f_1(\mathbf{U})+\sum_v\Big[\langle\mathbf{P}^v,\mathbf{\Gamma}^v\rangle+\frac{\rho}{2}||\mathbf{\Gamma}^v||_F^2\Big]\\
&=\arg\min_{\mathbf{U}}\sum_{i=1}^m\Big\{\frac{1}{2m}(y_i-\sum_{v=1}^s\mathbf{x}_i^{v\top}\mathbf{u}_i^v)^2+\sum_v(\mathbf{P}_{i\cdot}^v\boldsymbol{\gamma}_i^v\\
&\quad+\frac{\rho}{2}||\boldsymbol{\gamma}_i^v||^2)\Big\}
\end{aligned}$$

where $\mathbf{P}_{i\cdot}^v\in\mathbb{R}^{1\times n_v}$ is the row vector of $\mathbf{P}^v$. Obviously, the problem is decomposable for many variables $\{\mathbf{U}_i\}$. The decomposed subproblem with respect to $\mathbf{U}_i$ is

$$\begin{aligned}
&\arg\min_{\mathbf{U}_i}\frac{1}{2m}(y_i-\sum_{v=1}^s\mathbf{x}_i^{v\top}\mathbf{u}_i^v)^2+\sum_v(\mathbf{P}_{i\cdot}^v\boldsymbol{\gamma}_i^v+\frac{\rho}{2}||\boldsymbol{\gamma}_i^v||^2),\\
&=\arg\min_{\mathbf{U}_i}\frac{1}{2m}(y_i-\mathbf{x}_i{}^\top\mathbf{U}_i)^2+\mathbf{P}_i^\top(\mathbf{C}_i-\mathbf{U}_i)\\
&\quad+\frac{\rho}{2}||\mathbf{C}_i-\mathbf{U}_i||^2,
\end{aligned}\tag{12}$$

where $\mathbf{x}_i=(x_i^{1\top},...,x_i^{s\top})^\top\in\mathbb{R}^N$, $\mathbf{P}_i=(\mathbf{P}_{i\cdot}^1,...,\mathbf{P}_{i\cdot}^s)^\top\in\mathbb{R}^N$, and $\mathbf{C}_i=(w_i^1\boldsymbol{\beta}^{1\top},...,w_i^s\boldsymbol{\beta}^{s\top})^\top\in\mathbb{R}^N$. Eq.(12) can be proved to be a convex problem with respect to $\mathbf{U}_i$. So it can easily be solved by gradient descent.

4) **the solution of J**
Inspired by thresholding analysis [45], we solve $\mathbf{J}$ as

$$\begin{aligned}
\mathbf{J}&\leftarrow\arg\min_{\mathbf{J}}\lambda_R||\mathbf{J}||_*+\langle\mathbf{R},\mathbf{J}-\mathbf{W}\rangle+\frac{\xi}{2}||\mathbf{J}-\mathbf{W}||_F^2,\\
&=\arg\min_{\mathbf{J}}\frac{\lambda_R}{\xi}||\mathbf{J}||_*+\frac{1}{2}||\mathbf{J}-(\mathbf{W}-\mathbf{R}/\xi)||_F^2,\\
&=\mathbf{X}\mathcal{S}_\eta[\mathbf{S}]\mathbf{Y}^\top
\end{aligned}\tag{13}$$

where $\eta=\lambda_R/\xi$, $\mathbf{XSY}^\top$ is the SVD of $\mathbf{W}-\mathbf{R}/\xi$.

C. Summarization of Algorithm

Algorithm 1 shows the working process of our method. In the procedure of alternating optimization, each step of optimization is convex, whose solution decreases the original objective in Eq.(7). Moreover, the augmented Lagrange terms in Eq.(8) can improve the convergence rate of multiple variables. So the procedure in Algorithm 1 will converge fast.

As shown in Algorithm 1, in each iteration, the computational complexity for $\{\mathbf{w}^v\}$, $\{\boldsymbol{\beta}^v\}$, $\{\mathbf{U}_i\}$ and $\mathbf{J}$ ($v=1,...,s,\ i=1,...,m$) are $\mathcal{O}(mn+ms)$, $\mathcal{O}(mn)$, $\mathcal{O}(mn)$ and $\mathcal{O}(ms^2)$, respectively. Thus, the total complexity is $\mathcal{O}(Tm(n+s^2))$, where T is the number of iterations. Note that in most of our cases $n>>s$, so Algorithm 1 is mainly computed in the time of $\mathcal{O}(Tmn)$. Moreover, the proposed algorithm can be easily parallelized for efficient computation. (Please refer to Lines 3-9 in Algorithm 1)

D. Classifier-Level Fusion for classification

After obtaining the $\boldsymbol{\beta}^v$ of each view, the features with positive values will be selected. With the newly selected features, we learn view-specific classifiers for each view. The prediction results of these classifiers on testing data are then fused in the following two ways.

(1) Majority voting. We count the prediction labels of all classifiers, and the class with the majority voting is viewed as the overall prediction. When two or more majority classes exist with equivalent voting, the class is selected randomly from these majority voting classes. For binary classification, their class labels are set to -1 and 1, respectively. Let $\mathcal{Z}$ be a multi-view testing sample preprocessed by MRM-Lasso, $\mathbf{z}^v$ is its view-specific pattern of the v^{th} view. The majority voting of $\mathcal{Z}$ is formulated as

$$\phi(\mathcal{Z})=\text{sgn}(\sum_{v=1}^s\phi^v(\mathbf{z}^v)),\tag{14}$$

Algorithm 1 MRM-Lasso via the ADMM alternating optimization method

Input: Training data $\{(\mathbf{x}_i^1, ..., \mathbf{x}_i^s, y_i)\}_{i=1}^m$ of s views, parameters $\lambda_S,\ \lambda_R$

Output: Weight matrix $\mathbf{W}$ and feature selection parameters $\{\boldsymbol{\beta}^v\}$

Initialize $\{\mathbf{P}^v\}_0, \mathbf{Q}_0, \mathbf{R}_0,\ \rho_0 > 0,\ \mu_0 > 0,\ \xi_0 > 0$, and $t = 0$.
while not converged **do**
  for $v \leftarrow \{1, ..., s\}$ **do**
    $(\mathbf{w}^v)_{t+1} \leftarrow$ solution by Eq.(10).
    $(\boldsymbol{\beta}^v)_{t+1} \leftarrow$ solution by Eq.(11).
  end for
  for $i \leftarrow \{1, ..., m\}$ **do**
    $(\mathbf{U}_i)_{t+1} \leftarrow$ solution by Eq.(12).
  end for
  $\mathbf{J}_{t+1} \leftarrow$ solution by Eq.(13).
  Update $\{\mathbf{P}^v\}_t, \mathbf{Q}_t, \mathbf{R}_t$ to $\{\mathbf{P}^v\}_{t+1}, \mathbf{Q}_{t+1}, \mathbf{R}_{t+1}$.
  Update ρ_t, μ_t, ξ_t to $\rho_{t+1}, \mu_{t+1}, \xi_{t+1}$.
end while
$\mathbf{W} = \mathbf{W}_{t+1}$, $\boldsymbol{\beta}^v = \boldsymbol{\beta}^v_{t+1}$.

where sgn is the symbolic function, ϕ^v is the v^{th} view classifier and $\phi^v(\mathbf{z}^v)$ is its prediction label for $\mathbf{z}^v$. When the value of $\phi(\mathcal{Z})$ is equivalent to zero, the class label is randomly predicted as 1 or -1. The majority voting of classifiers is abbreviated as 'Voting' in the following experiments.

(2) Weight fusion. Based on the learned weight matrix $\mathbf{W}$ by Algorithm 1, we compute the weight b^v of every view. It is evaluated as $b^v = \frac{\sum_{i=1}^m (\mathbf{W})_{iv}}{\sum_{i=1}^m \sum_{v'=1}^s (\mathbf{W})_{iv'}}$. Thus, the label of $\mathcal{Z}$ can be predicted as

$$\phi(\mathcal{Z}) = \text{sgn}(\sum_{v=1}^{s} b^v \cdot \phi^v(\mathbf{z}^v)). \tag{15}$$

The weight fusion of classifiers is abbreviated as 'Fusion' in the experiments. Thus, our method has two forms: '**MRM-Lasso Voting**' and '**MRM-Lasso Fusion**'.

V. Experiments on Classification Performance of MRM-Lasso

In this section, we evaluate the classification performance of our MRM-Lasso on four real-life datasets: Colon Cancer, Ads, WebKB Course and IXMAS, compared with five baseline algorithms. Then, the features selected by different algorithms are further investigated by changing the number of selected features.

A. Datasets

Colon Cancer [46]: This is a 2000-gene expression dataset, which includes 62 samples collected from colon cancer patients, where 40 are tumor biopsies and 22 are normal. Each sample contains 2000-dimensional continuous features. The Colon Cancer dataset (**Colon** for short) is widely used for feature selection [31]. To evaluate the performance of our method, we simulate the dataset as a multi-view dataset by randomly partitioning it into three views.

Ads[3]: This describes a set of possible advertisement images on the web and the task is to predict whether an image is an advertisement or not. The dataset consists of multiple features, where each type of features is considered as a single view. In our experiments, we utilize three views (refer to article [5]), including image URL view (related to the image server name), destination URL view (related to the image URL), and alt view (related to alternate words in the HTML image tag).

WebKB Course[4]: This includes a collection of web pages from the computer science departments of four universities. WebKB Course (**Course** for short) has been extensively used for multi-view learning [2]. The web pages have two categories, course and non-course. Each web page has two views of representations: page view (the text content of the web page) and link view (the anchor text whose links point to the page). To distinguish course pages from non-course ones, we extract bag-of-words feature vectors via document tokenization[5] and generate the TFIDF feature vectors for both views.

IXMAS[1]: This is a multi-view video dataset for human activity recognition which consists of continuous movement videos captured by five individual cameras from five different angles. The dataset derives from 13 different actions of 12 actors. In this experiment, we select four couples of actions for classification: Scratch Head and Wave, Turn Around and Walk, Punch and Point, Check Watch and Cross Arms, which can be abbreviated as **ISW**, **ITW**, **IPP** and **ICC**, respectively. All five cameras are viewed as five different views, and the videos (or image sequences) taken by these cameras are the multi-view data in the experiments. For each frame image, HoG descriptors and silhouettes are extracted in a normalized bounding box in order to capture the global shape of images, and gradient-based descriptors are utilized to extract local motion features. The three kinds of features extracted from the image sequences are spliced to a high-dimensional feature to describe a persistent action. Details of the four datasets are given in Table III.

TABLE III
Details of the four datasets

Datasets	Different views		#Samples
	Name	#Features	
Colon	View1	500	62
	View2	700	
	View3	800	
Ads	img url	457	3279
	dest url	472	
	alt	111	
Course	page	3000	1051
	link	1747	
ISW/ITW/IPP/ICC	Camera1	960	75/76/74/74
	Camera2	960	
	Camera3	960	
	Camera4	960	
	Camera5	960	

B. Experimental Settings

Since feature selection is a key preprocessing step, the performance of feature selection methods has a significant impact on the following classification. For the classifier adopted in this paper, we employ the SVM with RBF kernel

[3]UCI data: http://archive.ics.uci.edu/ml/datasets/Internet+Advertisements
[4]http://www.cs.cmu.edu/afs/cs/project/theo-11/www/wwkb/
[5]Bow toolkit: http://www.cs.cmu.edu/ mccallum/bow

(RBF-SVM), an efficient and popular classifier used in many studies. To evaluate the efficiency of our proposed methods as well as to compare them with the baselines, three steps must be undertaken: (1) multi-view feature selection by feature selection methods; (2) for each view, training RBF-SVM classifiers respectively on the data after feature selection; and (3) majority voting in predicting results of every view to generate the overall prediction in the testing data. For the parameters in SVM learner (e.g., C and γ), we perform the inner 10-fold cross validation on the whole training data (without feature selection) for every view, which can quickly choose view-specific parameters and avoid the time-consuming parameter search in the experimental evaluation.

The comparison experiments of the feature selection baselines for classification performance are presented on four real-life datasets. **Accuracy** and **F1 score**[6] are adopted as the evaluation metrics. **Accuracy** is defined as the number of correctly classified samples divided by the total of testing samples. **F1 score** is the harmonic mean of **Precision** and **Recall** for each class. For binary classification in our experiments, we measure the positive F1-score for the positive class and the negative F1-score for the negative class. They are abbreviated as **pos-F1** and **neg-F1**, respectively, in the following sections. To improve the reliability of the classification results, we generate 100 random partitionings of all the four datasets into training and testing data (the fraction of training data is set as 75%). 100 classification results are generated and evaluated by their mean and variance on **accuracy** and **F1 score**. The parameters λ_S and λ_R in MRM-Lasso are learned via 10-fold cross validation. Optimal values of the two parameters are grid-searched from 10^i ($i = $ -5, -4, ..., 5).

C. Comparisons with Other Dimension Reduction Methods

To validate the superiority of our two algorithms, we compare them with the six baseline algorithms, i.e. Unselect, PCA, Lasso [19], mRMR [34], Feng's [14] and Tang's [15]. Unselect is the simplest method by which the whole data is utilized to learn the SVM classification without feature selection; PCA is one of the simplest but most popularly used dimension reduction algorithms; Lasso is the sparse feature selection method with L_1-norm constraint; mRMR is a popular feature selection method, which measures mutual information to find the most relevant features w.r.t. class labels. Feng's and Tang's methods are unsupervised multi-view feature selection for clustering. To compare them with MRM-Lasso, we simply evaluate their selected features without considering their clustering performance.

The experimental results (the mean and variance of accuracy, neg-F1 and pos-F1) of seven algorithms on seven groups of data are listed in Table IV, where four groups of data ISW, ITW, IPP and ICC are generated from IXMAS. For different algorithms, the highest mean and the lowest variance on Accuracy, pos-F1 and neg-F1 are in bold.

(1) With respect to the performance of different algorithms, the results in Table IV depict that compared to the baselines, our two algorithms achieve the best performance with the highest values of accuracy, positive F1 and negative F1 on all seven types of data, which demonstrates the strong discrimination capability of our selected features. In relation to our two algorithms, MRM-Lasso Fusion is superior to MRM-Lasso Voting on four types of data (Course, ITW, IPP and ICC). In the other data, MRM-Lasso Fusion is comparable with MRM-Lasso Voting. The reason is that the learned weights in the classifier-level fusion phase are helpful to improve the fusion results of multiple classifiers. The Unselect method performs the worst on six types of data (except for Ads), because a lot of redundant and irrelevant features greatly hinder the learning. Compared to PCA, Lasso and mRMR, our two algorithms outperform them, which indicates that cross-view low-rank constraints can enhance the performance of multi-view feature selection. Also, compared to Feng's and Tang's algorithms, the better performance of our two algorithms discloses the efficiency of class label information for selecting discriminative features.

(2) With respect to the performance on different datasets, we can see that our two algorithms achieve good performance on all seven types of data (including gene, text and video data). In contrast, the baselines only perform well on one or two types of data (e.g., mRMR prefers the genetic data in the Colon and video data in the ICC, while Lasso is outstanding for the ISW data), which suggests that our methods have more stable performance against different data types. Note that applying the feature selection strategy to the Ads data is unnecessary, according to our experimental validation, because the Ads data can be well represented by the original feature space, and almost all the features are helpful for classification. So, the results of our method on the Ads data are indistinctive compared with Unselect and the feature selection baselines, but they are comparable with the baselines.

To statistically analyze the classification performance, the paired t-test values comparing MRM-Lasso with the baseline algorithms are computed from the three evaluation indices: Accuracy, pos-F1 and neg-F1. When the P-Value is less than 0.05, our method is significantly different from the corresponding baseline. Table V shows the P-Value results on the Colon, Course, ISW, ITW, IPP and ICC. It shows that our algorithms result in a significantly improved performance on most of the data. Though the performance of MRM-Lasso Voting on IPP data is slightly superior to the baselines w.r.t. neg-F1, MRM-Lasso Fusion is more outstanding for the baselines than MRM-Lasso Voting, which validates the significance of the proposed pattern weights again.

D. Comparisons with Other Multi-Kernel / Multi-View Methods

In this subsection, we evaluate the performance of two multiple kernel learning algorithms (L_2-MKL SVM and L_∞-MKL SVM [47]), a multi-view subspace learning algorithm - Multi-view Fisher Discriminative Analysis (MFDA) [48], and a multi-modality recognition algorithm - Sparse Multimodal Biometrics Recognition (SMBR) [49] for comparison with our methods. The comparison results are shown in Table V. In Table V, we can see that L_2-MKL SVM achieves better performance than L_∞-MKL SVM, and SMBR performs better than MFDA on almost all datasets expect the Colon data. Also, all of them are much worse than our proposed algorithms on all five data sets excluding IPP, on which L_2-MKL SVM performs slightly better than MRM-Lasso

[6]F1 score is also called F-measure in other articles.

TABLE IV
THE CLASSIFICATION RESULTS (MEAN AND VARIANCE IN PERCENTAGE) OF SVM WITH SEVEN DIFFERENT ALGORITHMS ON SEVEN GROUPS OF DATA. IN THESE ALGORITHMS, THE HIGHEST MEAN AND THE LOWEST VARIANCE ON ACCURACY, POS-F1 AND NEG-F1 ARE IN BOLD. WEBKB COURSE DATASET IS SHORT FOR COURSE. THE DATA ABOUT SCRATCH HEAD AND WAVE, TURN AROUND AND WALK, PUNCH AND POINT, CHECK WATCH AND CROSS ARMS ON IXMAS DATASET CAN BE ABBREVIATED AS ISW, ITW, IPP AND ICC, RESPECTIVELY.

Data	Evaluation	Unselect	PCA	Lasso	mRMR	Feng's [14]	Tang's [15]	MRM-Lasso Voting / Fusion
Colon	**Accuracy**	86.81 ± **0.37**	87.69 ± 0.60	87.44 ± 0.49	88.38 ± 0.56	87.63 ± 0.49	87.88 ± 0.46	**89.64** ± 0.44 / **89.64** ± 0.44
	neg-F1	89.34 ± **0.25**	90.00 ± 0.40	89.82 ± 0.34	90.55 ± 0.38	89.83 ± 0.36	90.46 ± 0.29	**91.64** ± 0.30 / **91.64** ± 0.30
	pos-F1	82.33 ± **0.71**	83.64 ± 1.09	83.27 ± 0.90	84.63 ± 1.01	83.80 ± 0.85	82.99 ± 1.03	**86.34** ± 0.80 / **86.34** ± 0.80
Ads	**Accuracy**	94.83 ± 0.00	85.98 ± 0.00	94.96 ± 0.00	95.00 ± 0.00	94.75 ± 0.00	94.97 ± 0.00	**95.04** ± 0.00 / **95.04** ± 0.00
	neg-F1	97.07 ± 0.00	92.46 ± 0.00	97.15 ± 0.00	97.17 ± 0.00	97.03 ± 0.00	97.15 ± 0.00	**97.19** ± 0.00 / **97.19** ± 0.00
	pos-F1	77.66 ± 0.09	0.00 ± 0.00	78.27 ± 0.09	78.50 ± 0.10	77.23 ± 0.09	78.35 ± 0.07	**78.66** ± 0.10 / **78.66** ± 0.10
Course	**Accuracy**	91.97 ± 0.16	90.83 ± 0.02	93.34 ± **0.01**	95.66 ± **0.01**	94.28 ± **0.01**	94.83 ± **0.01**	96.59 ± **0.01** / **96.81** ± **0.01**
	neg-F1	95.10 ± 0.01	94.32 ± 0.01	95.88 ± 0.01	97.27 ± **0.00**	96.42 ± 0.01	96.76 ± 0.01	97.85 ± **0.00** / **97.99** ± **0.00**
	pos-F1	77.75 ± 0.17	76.18 ± 0.14	82.63 ± 0.13	89.32 ± 0.08	85.66 ± 0.10	87.09 ± 0.09	91.71 ± **0.07** / **92.21** ± **0.07**
ISW	**Accuracy**	83.45 ± 0.49	89.03 ± 0.57	88.72 ± 0.38	87.34 ± 0.46	85.15 ± 0.55	84.77 ± 0.58	**92.88** ± **0.26** / **92.88** ± **0.26**
	neg-F1	80.40 ± 1.04	88.59 ± 0.70	88.19 ± 0.35	87.52 ± 0.44	85.75 ± 0.49	85.04 ± 0.56	**92.90** ± **0.26** / **92.90** ± **0.26**
	pos-F1	85.39 ± 0.35	89.23 ± 0.53	88.08 ± 0.44	86.91 ± 0.57	84.13 ± 0.73	84.07 ± 0.76	**92.76** ± **0.26** / **92.76** ± **0.26**
ITW	**Accuracy**	92.80 ± 0.37	94.93 ± 0.31	93.68 ± 0.31	94.16 ± 0.32	93.46 ± 0.41	94.55 ± 0.25	96.29 ± 0.23 / **96.34** ± **0.22**
	neg-F1	91.82 ± 0.59	95.41 ± 0.24	94.20 ± 0.25	94.69 ± 0.25	93.91 ± 0.34	95.00 ± 0.20	96.54 ± **0.19** / **96.58** ± **0.19**
	pos-F1	93.50 ± **0.26**	94.30 ± 0.42	92.99 ± 0.42	93.47 ± 0.44	92.86 ± 0.53	93.97 ± 0.34	95.97 ± 0.29 / **96.03** ± 0.28
IPP	**Accuracy**	87.36 ± **0.42**	86.20 ± 0.67	87.64 ± 0.61	88.48 ± 0.43	87.62 ± 0.46	88.30 ± 0.47	89.89 ± 0.55 / **92.19** ± **0.42**
	neg-F1	88.88 ± **0.29**	87.23 ± 0.63	88.68 ± 0.52	88.66 ± 0.49	88.49 ± 0.39	89.30 ± 0.39	90.25 ± 0.55 / **92.66** ± 0.37
	pos-F1	85.06 ± 0.75	84.59 ± 0.88	86.17 ± 0.81	87.95 ± 0.48	86.34 ± 0.64	86.87 ± 0.67	89.32 ± 0.62 / **91.52** ± **0.53**
ICC	**Accuracy**	83.94 ± 0.67	85.67 ± 0.59	88.24 ± 0.69	89.04 ± **0.32**	84.33 ± 0.66	86.75 ± 0.53	90.99 ± 0.34 / **91.12** ± 0.39
	neg-F1	80.53 ± 1.29	84.18 ± 0.90	87.98 ± 0.71	87.39 ± 0.53	82.91 ± 0.90	85.56 ± 0.72	90.22 ± **0.43** / **90.63** ± **0.43**
	pos-F1	86.06 ± 0.46	86.65 ± 0.46	88.28 ± 0.75	90.18 ± **0.23**	85.27 ± 0.57	87.52 ± 0.47	**91.53** ± 0.29 / 91.48 ± 0.36

TABLE VI
THE CLASSIFICATION RESULTS (IN PERCENTAGE) WITH FOUR DIFFERENT MULTI-KERNEL/MULTI-VIEW/MULTI-MODALITY ALGORITHMS ON FIVE GROUPS OF DATA. IN THESE ALGORITHMS, THE HIGHEST MEAN AND THE LOWEST VARIANCE ON ACCURACY, POS-F1 AND NEG-F1 ARE IN BOLD.

Data	Evaluation	L_2-MKL SVM [47]	L_∞-MKL SVM [47]	MFDA [48]	SMBR [49]	MRM-Lasso Voting / Fusion
Colon	**Accuracy(mean±var)**	85.19 ± 0.57	84.50 ± 0.58	84.38 ± 0.68	83.00 ± 0.68	**89.64** ± 0.44 / **89.64** ± **0.44**
	neg-F1(mean±var)	87.98 ± 0.39	86.79 ± 0.54	87.06 ± 0.53	86.80 ± 0.41	**91.64** ± **0.30** / **91.64** ± **0.30**
	pos-F1(mean±var)	80.01 ± 1.28	80.60 ± **0.77**	79.78 ± 1.10	75.43 ± 1.73	**86.34** ± 0.80 / **86.34** ± 0.80
ISW	**Accuracy(mean±var)**	91.97 ± 0.35	84.57 ± 0.60	85.52 ± 0.55	88.90 ± 0.36	**92.88** ± **0.26** / **92.88** ± **0.26**
	neg-F1(mean±var)	91.67 ± 0.39	84.61 ± 1.02	84.58 ± 0.72	89.17 ± 0.37	**92.90** ± **0.26** / **92.90** ± **0.26**
	pos-F1(mean±var)	92.11 ± 0.35	83.41 ± 0.82	85.99 ± 0.55	88.39 ± 0.44	**92.76** ± **0.26** / **92.76** ± **0.26**
ITW	**Accuracy(mean±var)**	92.61 ± 0.44	89.89 ± 0.47	92.29 ± 0.47	92.89 ± 0.25	96.29 ± 0.23 / **96.34** ± **0.22**
	neg-F1(mean±var)	92.78 ± 0.42	88.63 ± 0.79	91.64 ± 0.62	93.53 ± 0.20	96.54 ± **0.19** / **96.58** ± **0.19**
	pos-F1(mean±var)	92.28 ± 0.50	90.63 ± 0.39	92.68 ± 0.43	92.07 ± 0.33	95.97 ± 0.29 / **96.03** ± **0.28**
IPP	**Accuracy(mean±var)**	90.31 ± 0.47	88.63 ± 0.58	83.42 ± 0.68	87.22 ± 0.58	89.89 ± 0.55 / **92.19** ± **0.42**
	neg-F1(mean±var)	91.05 ± 0.41	90.30 ± **0.37**	82.90 ± 0.87	88.06 ± 0.49	90.25 ± 0.55 / **92.66** ± **0.37**
	pos-F1(mean±var)	89.30 ± 0.59	86.01 ± 1.12	83.57 ± 0.64	86.02 ± 0.78	89.32 ± 0.62 / **91.52** ± **0.53**
ICC	**Accuracy(mean±var)**	87.54 ± 0.55	79.56 ± 1.04	75.66 ± 1.07	87.98 ± 0.54	90.99 ± **0.34** / **91.12** ± 0.39
	neg-F1(mean±var)	86.36 ± 0.74	71.19 ± 3.45	75.88 ± 1.08	86.97 ± 0.70	90.22 ± **0.43** / **90.63** ± **0.43**
	pos-F1(mean±var)	88.34 ± 0.47	83.85 ± 0.47	74.47 ± 1.59	88.59 ± 0.50	**91.53** ± **0.29** / 91.48 ± 0.36

Voting but worse than MRM-Lasso Fusion. It shows that the proposed algorithms have superior performance, compared to these multi-kernel / multi-view / multi-modality algorithms, which demonstrates that feature selection in multi-view data is helpful for performance improvement.

E. Evaluation on a Different Number of Selected Features

To further demonstrate the strong discrimination of the selected features with a different number of selected features, we change the number of selected features in every view fifteen times, and then evaluate the corresponding classification performance of our MRM-Lasso on the Colon data, compared to mRMR and Lasso. We add $13, 17$ and 22 new features from three views, respectively, into the feature set each round. The results for multi-view classification and every single-view classification are depicted in Fig. 3. (1) It is observed that when the total of the selected features ranges from 364 to 780, our algorithm, which selects the equivalent number of features to mRMR and Lasso, outperforms them in terms of higher accuracy in multi-view, view 1 and view 2, respectively (view 3 is comparable). (2) Fig. 3 also discloses that when achieving equivalent classification performance, our MRM-Lasso requires fewer features compared with mRMR and Lasso.

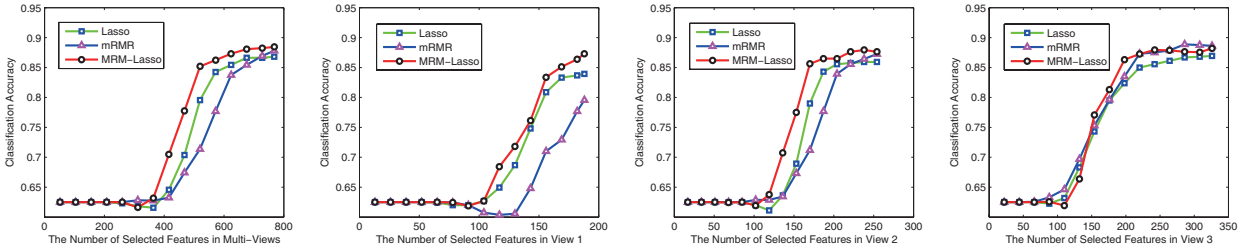
VI. EXPERIMENTS ON UNIQUENESS OF MRM-LASSO

In this section, we present experiments to exploit the significance of pattern weighting for multi-view feature selection and evaluate the effects of the two parameters λ_S, λ_R in MRM-Lasso on weight matrix $\mathbf{W}$ and feature selection vectors $\{\boldsymbol{\beta}^v\}$. We conduct experiments on the Colon and IXMAS datasets for demonstration.

A. View-Weighting Lasso

To validate the efficiency of pattern-specific weights in MRM-Lasso, we compare our method with traditional Lasso (without any weights) as well as view-weighting Lasso (**View-Lasso** for short). The objective of view-weighting Lasso is formulated as

$$\min_{\alpha^v,\boldsymbol{\beta}^v} \sum_{i=1}^{m}(y_i - \sum_{v=1}^{s}\alpha^v \mathbf{x}_i^{v\top}\boldsymbol{\beta}^v)^2 + \lambda_s \sum_{i=1}^{s}||\boldsymbol{\beta}^v||_1$$
$$s.t. \sum_{v=1}^{s}\alpha^v = 1,\ \alpha^v > 0,$$



(a) Multi-Views (b) View 1 (c) View 2 (d) View 3

Fig. 3. The SVM classification accuracy in different views along with the increasing number of selected features on the Colon dataset

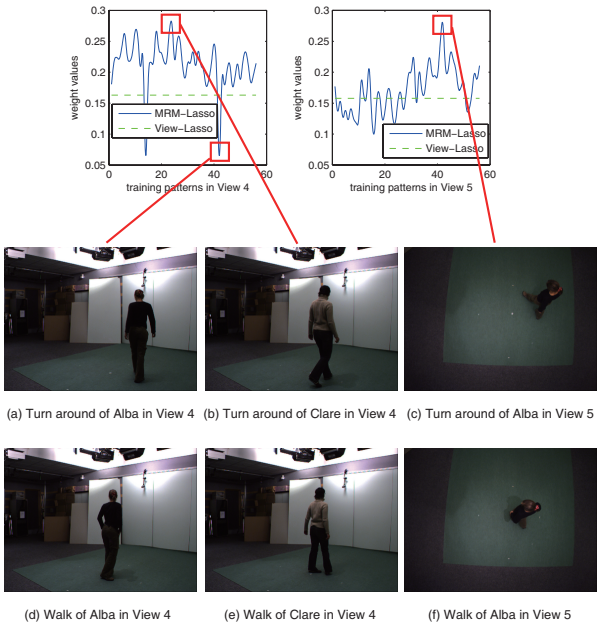
TABLE V
THE P-VALUE RESULTS OF CLASSIFICATION PERFORMANCE OF MRM-LASSO AND THE BASELINES ON COLON, COURSE, ISW, ITW, IPP AND ICC. '–' REPRESENTS OUR METHOD SIGNIFICANTLY IMPROVES PERFORMANCE COMPARED WITH THE CORRESPONDING BASELINE.

Data	Methods	MRM-Lasso Voting / Fusion		
		Accuracy	neg-F1	pos-F1
Colon	Unselect	– / –	– / –	– / –
	Lasso	– / –	– / –	– / –
	mRMR	0.18 / 0.18	0.18 / 0.18	0.19 / 0.19
	Feng's [14]	– / –	– / –	0.06 / 0.06
	Tang's [15]	– / –	0.08 / 0.08	– / –
Course	Unselect	– / –	– / –	– / –
	Lasso	– / –	– / –	– / –
	mRMR	– / –	– / –	– / –
	Feng's [14]	– / –	– / –	– / –
	Tang's [15]	– / –	– / –	– / –
ISW	Unselect	– / –	– / –	– / –
	Lasso	– / –	– / –	– / –
	mRMR	– / –	– / –	– / –
	Feng's [14]	– / –	– / –	– / –
	Tang's [15]	– / –	– / –	– / –
ITW	Unselect	– / –	– / –	– / –
	Lasso	– / –	– / –	– / –
	mRMR	– / –	– / –	– / –
	Feng's [14]	– / –	– / –	– / –
	Tang's [15]	– / –	– / –	– / –
IPP	Unselect	– / –	0.16 / –	– / –
	Lasso	– / –	0.14 / –	– / –
	mRMR	0.18 / –	0.14 / –	0.21 / –
	Feng's [14]	– / –	0.08 / –	– / –
	Tang's [15]	0.11 / –	0.32 / –	– / –
ICC	Unselect	– / –	– / –	– / –
	Lasso	– / –	0.05 / –	– / –
	mRMR	– / –	– / –	0.06 / 0.07
	Feng's [14]	– / –	– / –	– / –
	Tang's [15]	– / –	– / –	– / –

where $\alpha^v \in \mathbb{R}$ is the weight of the v^{th} view. Local optimal solutions can be easily found by ADMM. Due to similar optimization with that of MRM-Lasso (see Section IV-B), we omit the optimization details of View-Lasso.

B. Comparison of View-Lasso and MRM-Lasso

For View-Lasso (view-level) and MRM-Lasso (sample-level), we investigate the distribution of their weights in different views. We assume that all the pattern weights of View-Lasso in a particular view are the same, which is equivalent to the learned view weight. We repeat the two algorithms 100 times and compute the average values of each pattern since the solutions of both View-Lasso and MRM-Lasso are local optimal. For **ITW** data, which describes 'Turn around' against 'Walk', the average pattern weights, generated by both View-Lasso and MRM-Lasso, are shown in Fig. 4. It is observed that pattern weights change along with different action samples derived from different actors. To illustrate the significance of pattern weights, we take three locations in Fig. 4 for example (see Fig. 5).



(a) Turn around of Alba in View 4 (b) Turn around of Clare in View 4 (c) Turn around of Alba in View 5

(d) Walk of Alba in View 4 (e) Walk of Clare in View 4 (f) Walk of Alba in View 5

Fig. 5. An example of Turn around vs. Walk in View 4 and View 5.

• **Pattern weights of a particular sample across views**

In Fig. 5, the trough in View 4 (at Location 42) and the peak in View 5 (at Location 42) correspond to Alba's action 'Turn around' in View 4 (see (a)) and View 5 (see (c)), respectively. Comparing (a)(d) and (c)(f) respectively, we found that View 5 has a stronger recognition capability for the 'Turn around' of Alba than View 4 because of her obvious strides shown in View 5. So, we can see that for a particular sample across different views, the action pattern with the higher discriminative capability generally has the higher pattern weight learned by MRM-Lasso.

• **Pattern weights within a certain view**

In Fig. 5, the peak (at Location 24) in View 4 refers to Clare's 'Turn around' (see (b)), which is more significant than Alba's 'Turn around' (see (a)). From (a), (b), (d) and (e), we can see that the movements of Clare are more discriminative and significant than Alba's. Thus, it can be observed that for different patterns within a particular view, the pattern with the higher discriminative capability usually has the higher pattern weight learned by MRM-Lasso.

The above example suggests that MRM-Lasso (sample-level) can explore the view-specific and pattern-specific discriminations of multi-view data via pattern weights, which is significant for real-life complex data. However,

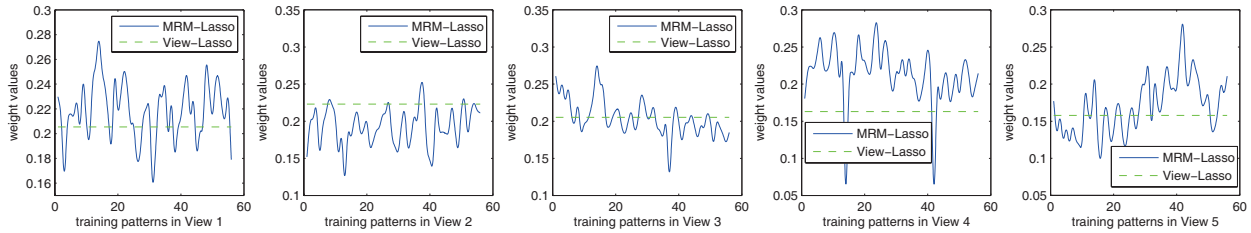


Fig. 4. The distribution of pattern weights in different views on ITW data

TABLE VII
THE CLASSIFICATION RESULTS (MEAN AND VARIANCE IN PERCENTAGE) OF SVM WITH DIFFERENT WEIGHTING METHODS. IN THESE METHODS, THE HIGHEST MEAN AND THE LOWEST VARIANCE ON ACCURACY, POS-F1 AND NEG-F1 ARE IN BOLD.

Data	Evaluation	Lasso	View-Lasso	MRM-Lasso
Colon	**Accuracy**	87.44 ± 0.49	88.31 ± 0.46	$\mathbf{89.64 \pm 0.44}$
	neg-F1	89.82 ± 0.34	90.68 ± 0.30	$\mathbf{91.64 \pm 0.30}$
	pos-F1	83.27 ± 0.90	84.07 ± 0.88	$\mathbf{86.34 \pm 0.80}$

the weight of View-Lasso (view-level) cannot provide the pattern information of multi-view.

The comparison results of the classification performance on the Colon data are shown in Table VII. It is obvious that our MRM-Lasso performs the best, which suggests that sample-level pattern weights are significant in the selection of discriminative features for classification.

C. The Effects of Parameters

In MRM-Lasso, parameters λ_S, λ_R are the parameters of L_1-sparsity term and low-rank term, respectively. To visually evaluate their effects on our method, we measure many different combinations of parameters λ_S and λ_R. Specifically, λ_S varies from $0, 0.05, ..., 0.05 \times n$ $(n = 0, ..., 10)$, and λ_R is changed from $0, 5, ..., 5 \times n$ $(n = 0, ..., 10)$.

- **pattern weights**

For different pairs of parameters, the average variances of pattern weights across views on the Colon data are computed, as shown in Fig. 6(a). As λ_R increases, the average variances decrease generally, which indicates that strong low-rank constraints can weaken the disagreement of patterns across views.

- **the number of selected features**

For different pairs of parameters, the average number of selected features across views on the Colon data is computed, as shown in Fig. 6(b). Obviously, the average number decreases as λ_S increases. Specifically, most features are selected when $\lambda_S = 0$. Fig. 6(b) suggests that the number of selection can be controlled by assigning a value to λ_S.

- **classification performance**

The classification voting accuracies with different pairs of parameters on the Colon data are depicted in Fig. 6(c). On the whole, the accuracy values, along with different pairs of parameters, have weak fluctuations with the highest value being 89.26% and the lowest value 87.12%. When $\lambda_S = 0$ or $\lambda_R = 0$, the classification results are undesirable with low values of accuracy, which indicates the importance of both sparse term and low-rank term in MRM-Lasso. When $\lambda_S = 0.35$ or $\lambda_R = 35$, MRM-Lasso achieves the best results with the highest accuracy 89.26%. Fig. 6(c) shows the low sensitivity of our method for parameters, as well as its superiority for classification tasks.

TABLE VIII
THE COMPARISON RESULTS (IN PERCENTAGE) OF GW-LP AND LW-KNN WITH LP AND KNN ON COLON, RESPECTIVELY.

Colon	LP	GW-LP	KNN	LW-KNN
Accuracy	77.81	81.25	77.19	87.50
neg-F1	82.57	85.71	83.30	90.00
pos-F1	69.39	72.73	63.65	83.33

D. The Effects of Pattern Weights for Classification

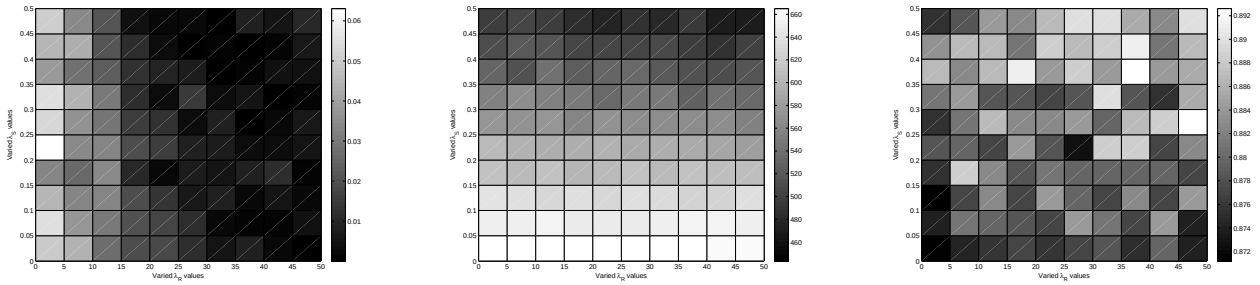
Since multi-view feature selection is the main problem that we emphasize, pattern weights are not directly employed at the classification stage. To explore the effects of pattern weights on classification, we engage in a discussion about generalizing the pattern weights from training data to testing data. Several information propagation methods can be adopted to transmit the weights of training data to testing data for pattern weight-based decision. To test the performance of pattern weights for classification, we design two simple pattern-weight-based classifiers as base classifiers of each view and then fuse the decision values via weighted fusion, as proposed in our paper. The two pattern-weight-based classifiers are introduced below.

- (Globally) Weighted Label Propagation algorithm (GW-LP) considers both the weights of training samples and their similarities with testing samples. For a certain view, the decision function of this base classifier can be written by $g(\mathbf{x}_t^v) = \text{sgn}(\sum_{i=1}^m S(\mathbf{x}_i^v, \mathbf{x}_t^v) w_i^v y_i)$, where $\mathbf{x}_i^v$ and $\mathbf{x}_t^v$ are a training sample and a testing sample in the current view respectively. w_i^v and y_i are the weight and the label of $\mathbf{x}_i^v$, respectively. $S(,)$ is the similarity function, which can be computed by the kernel function. sgn is the symbolic function.
- (Locally) Weighted KNN algorithm (LW-KNN) introduces the sample weights among K-nearest neighbors of testing samples to predict their labels. For a certain view, the decision function of this base classifier can be written by $g(\mathbf{x}_t^v) = \text{sgn}(\sum_{k \in N(\mathbf{x}_t^v)} w_k^v y_k)$, where $N(\mathbf{x}_t^v)$ represents the K nearest neighbors (from the training instances) of the testing instance $\mathbf{x}_t^v$ in the current view.

To evaluate the performance of pattern weights in GW-LP and LW-KNN, similarity-based label propagation (LP) and KNN (not including the pattern weights) are compared with GW-LP and LW-KNN, respectively. The comparison results on the Colon data are shown in Table VIII. We can see from Table VIII that GW-LP is better than LP, and LW-KNN is better than KNN, which shows the significance of pattern weights.

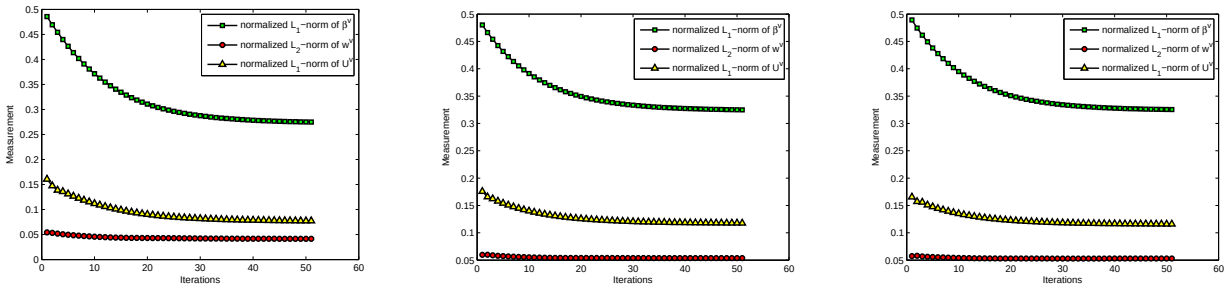
E. The Evaluation of the Connection between $\mathbf{W}$ and $\{\boldsymbol{\beta}^v\}$

In the proposed MRM-Lasso, both view significance and sample significance are measured by learning $\mathbf{W}$ with low-rank assumption, while feature significance is calculated by $\{\boldsymbol{\beta}^v\}$. To evaluate the connection between $\mathbf{W}$ and $\{\boldsymbol{\beta}^v\}$ to clarify the learning process, we add several descriptions and provide two additional experiments.



(a) The average variance of W across views (b) The average number of selected features across views (c) The average accuracy for SVM classification

Fig. 6. The average variances of W across views, the average feature selection number across views and the average accuracy for SVM classification against different pairs of parameter λ_S and λ_R on the Colon dataset.



(a) The changes of view 1 (b) The changes of view 2 (c) The changes of view 3

Fig. 7. The intra-view L_1/L_2-norm changes of $\mathbf{U}^v$, $\mathbf{w}^v$ and $\boldsymbol{\beta}^v$ in different iterations on Colon data

• Regarding how $\boldsymbol{\beta}^v$ is constrained, $\boldsymbol{\beta}^v$s are implicitly constrained by the corresponding $\mathbf{w}^v$ in the objective function. Specifically, in the optimization process, slack variables $\{\mathbf{U}^v\}$ are introduced to build an explicit connection between $\mathbf{w}^v$ and $\boldsymbol{\beta}^v$, where $\mathbf{U}^v = \mathbf{w}^v\boldsymbol{\beta}^{v\top}$. In each iteration, both the solutions of $\mathbf{w}^v$ and $\boldsymbol{\beta}^v$ are correlated to the current $\mathbf{U}^v$. To validate the correlation, the intra-view changes of $\mathbf{U}^v$, $\mathbf{w}^v$ and $\boldsymbol{\beta}^v$ during different iterations are measured on the Colon data (see Fig. 7). In Fig. 7, as the iteration increases, it can be seen that the three curves of $\mathbf{U}^v$, $\mathbf{w}^v$ and $\boldsymbol{\beta}^v$ are consistent with our assumption ($\mathbf{U}^v = \mathbf{w}^v\boldsymbol{\beta}^{v\top}$). Thus, the addition of $\mathbf{U}^v$ can help measure the changing relationship of $\mathbf{w}^v$ and $\boldsymbol{\beta}^v$ of different iterations and improve the search for the optimal $\boldsymbol{\beta}^v$.

• The advantage of using $\mathbf{W}$ to constrain $\boldsymbol{\beta}^v$ includes: (1) no direct constraints on inter-view $\boldsymbol{\beta}^v$s can help select view-specific significant features; and (2) a low-rank constraint on $\mathbf{W}$ can prevent $\{\boldsymbol{\beta}^v\}$ from falling into 'one view overpowers other views'. Of different views, we measure the variance of $\mathbf{U}^v$, $\mathbf{w}^v$ and $\boldsymbol{\beta}^v$ in different iterations (see Fig. 8). In Fig. 8, we can see that by imposing the low-rank constraint on $\mathbf{W}$, the average variance of $\mathbf{W}$ across views decreases dramatically, so that the inter-view variances of averaging vector $\boldsymbol{\beta}^v$ (see the bottom curve in Fig. 8) remain small and also stable during different iterations. Thus, $\boldsymbol{\beta}^v$ of a certain view would not overpower other views.

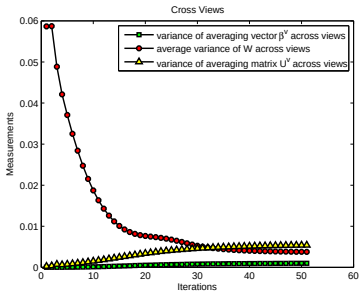


Fig. 8. The inter-view variance changes of $\mathbf{U}^v$, $\mathbf{w}^v$ and $\boldsymbol{\beta}^v$ in different iterations

VII. Conclusions

Specifically, this work proposes the MRM-Lasso method for sparse multi-view feature selection. Our method involves three highlights: (1) pattern-specific weights are introduced to evaluate the importance of patterns in a particular view; (2) a low-rank structure on the matrix consisting of the pattern-specific weights is constrained to efficiently capture the relevant patterns across views; and (3) Lasso is extended to multi-view scenarios to select the most discriminative features from multi-view simultaneously. The above feature selection coefficients and low-rank matrix are jointly learned by ADMM. Sufficient experiments demonstrate that our method outperforms the baseline algorithms on the Colon, Ads, WebKB Course and IXMAS datasets with the highest Accuracy and F1-score. Our method has more stable performance than the baselines against different data types (including videos, text, and genetic data). When changing the number of selected features, our method, which selects the equivalent number of features, has the highest accuracy compared with Lasso and mRMR. Moreover, as shown in comparison experiments with view-specific weights, it is demonstrated that the proposed pattern weights can explore the view-specific and pattern-specific discriminations of multi-view data, which is significant for real-life complex data. Thus, our future work is to further deal with incomplete multi-view data with missing patterns or noises via pattern weight learning.

References

[1] K. Chaudhuri, S. M. Kakade, K. Livescu, and S. Karthik, "Multi-view clustering via canonical correlation analysis," in *Proceedings of the 26th*

International Conference on Machine Learning, Montreal, Canada, Jun. 2009, pp. 129–136.

[2] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the 11th Annual Conference on Computational Learning Theory*, Madison, Wisconsin, Jul. 1998, pp. 92–100.

[3] J. D. R. Farquhar, D. R. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmak, "Two view learning: SVM-2K, theory and practice," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, B.C., Dec. 2005, pp. 355–362.

[4] R. K. Ando and T. Zhang, "Two-view feature generation model for semi-supervised learning," in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, Oregon, Jun. 2007, pp. 25–32.

[5] G. Li, K. Chang, and S. Hoi, "Multi-view semi-supervised learning with consensus," *IEEE Trans. Knowledge and Data Engineering*, vol. 24, pp. 2040–2051, Nov. 2012.

[6] X. Liu, L. De Lathauwer, S. Ji, W. Glanzel, and B. De Moor, "Multi-view partitioning via tensor methods," *IEEE Trans. Knowledge and Data Engineering*, vol. 25, pp. 1056–1069, May 2013.

[7] N. Chen, J. Zhu, F. Sun, and E. P. Xing, "Large-margin predictive latent subspace learning for multi-view data analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, pp. 2365–2378, Dec. 2012.

[8] D. P. Foster, R. Johnson, S. M. Kakade, and T. Zhang, "Multi-view dimensionality reduction via canonical correlation analysis," Toyota Technological Institute, Chicago, Illinois, Tech. Rep. TTI-TR-2008-4, Dec. 2008.

[9] M. White, Y. Yu, X. Zhang, and D. Schuurmans, "Convex multi-view subspace learning," in *Proc. Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, Dec. 2012, pp. 1682–1690.

[10] D. Zhai, H. Chang, S. Shan, X. Chen, and W. Gao, "Multiview metric learning with global consistency and local smoothness," *ACM Trans. on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, pp. 53:1–53:22, 2012.

[11] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 1, pp. 245–271, 1997.

[12] M. Ramona, G. Richard, and B. David, "Multiclass feature selection with kernel Gram-matrix-based criteria," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 23, pp. 1611–1623, Oct. 2012.

[13] M. Tan, I. W. Tsang, and L. Wang, "Minimax sparse logistic regression for very high-dimensional feature selection," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 24, pp. 1609–1622, Oct. 2013.

[14] Y. Feng, J. Xiao, Y. Zhuang, and X. Liu, "Adaptive unsupervised multi-view feature selection for visual concept recognition," in *Proceedings of the 11th Asian Conference on Computer Vision*, Daejeon, Korea, Nov. 2012, pp. 343–357.

[15] J. Tang, X. Hu, H. Gao, and H. Liu, "Unsupervised feature selection for multi-view data in social media," in *SIAM International Conference on Data Mining*, Austin, Texas, May 2013, pp. 270–278.

[16] H. Wang, F. Nie, and H. Huang, "Multi-view clustering and feature learning via structured sparsity," in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Jun. 2013, pp. 352–360.

[17] G. Ye, D. Liu, I. H. Jhuo, and S. F. Chang, "Robust late fusion with rank minimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, Jun. 2012, pp. 3021–3028.

[18] I. H. Jhuo, D. Liu, D. T. Lee, and S. F. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, Jun. 2012, pp. 2168–2175.

[19] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society (Series B)*, vol. 58, no. 1, pp. 267–288, 1996.

[20] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society (Series B)*, vol. 68, no. 1, pp. 49–67, 2006.

[21] J. Friedman, T. Hastie, and R. Tibshirani, "A note on the group Lasso and a sparse group Lasso," Jan. 2010, Available: http://arxiv.org/pdf/101.0736, preprint.

[22] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused Lasso," *Journal of the Royal Statistical Society (Series B)*, vol. 67, no. 1, pp. 91–108, 2005.

[23] G. Obozinski, B. Taskar, and M. I. Jordan, "Multi-task feature selection," Statistics Department, UC Berkeley, Berkeley, CA, Tech. Rep., Jun. 2006.

[24] G. Swirszcz and A. C. Lozano, "Multi-level Lasso for sparse multi-task regression," in *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, Jun. 2012, pp. 361–368.

[25] Y. Shi, S. Liao, Y. Gao, D. Zhang, Y. Gao, and D. Shen, "Prostate segmentation in CT images via spatial-constrained transductive Lasso," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, Oregon, Jun. 2013, pp. 2227–2234.

[26] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, Jun. 2011, pp. 625–632.

[27] A. Ahmed and E. P. Xing, "Recovering time-varying networks of dependencies in social and biological studies," *Proceedings of the National Academy of Sciences of the USA*, vol. 106, no. 29, pp. 11 878–11 883, 2009.

[28] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, no. 3, pp. 1157–1182, 2003.

[29] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1, pp. 273–324, 1997.

[30] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, B.C., Dec. 2005, pp. 507–514.

[31] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 389–422, 2002.

[32] I. Kononenko, "Estimating attributes: analysis and extensions of RELIEF," in *Proc. European Conference on Machine Learning*, Catania, Italy, Apr. 1994, pp. 171–182.

[33] Q. Gu, Z. Li, and J. Han, "Generalized Fisher score for feature selection," in *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, Barcelona, Spain, Jul. 2011, pp. 266–273.

[34] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, pp. 1226–1238, Aug. 2005.

[35] D. Li, N. Dimitrova, M. Li, and I. K. Sethi, "Multimedia content processing through cross-modal association," in *Proceedings of the 11th ACM International Conference on Multimedia*, Berkeley, CA, Nov. 2003, pp. 604–611.

[36] N. Quadrianto and C. H. Lampert, "Learning multi-view neighborhood preserving projections," in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, Jun. 2011, pp. 425–432.

[37] L. Ma, C. Wang, B. Xiao, and W. Zhou, "Sparse representation for face recognition based on discriminative low-rank dictionary learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, 2012, pp. 2586–2593.

[38] Y. Deng, Q. Dai, R. Liu, and Z. Zhang, "Low-rank structure learning via nonconvex heuristic recovery," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 24, pp. 383–396, Mar. 2013.

[39] J. Wright, Y. Peng, Y. Ma, A. Ganesh, and S. Rao, "Robust principal component analysis: exact recovery of corrupted low-rank matrices via convex optimization," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, B.C., Dec. 2009, pp. 2080–2088.

[40] G. Liu, Z. Lin, S. Yan, and J. Sun, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, pp. 171–184, Jan. 2013.

[41] D. Luo, F. Nie, C. Ding, and H. Huang, "Multi-subspace representation and discovery," in *Proc. ECML/PKDD*, Athens, Greece, Sep. 2011, pp. 405–420.

[42] Z. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL: CRC Press, 2012.

[43] Z. Wang, S. Chen, and T. Sun, "Multik-mhks: A novel multiple kernel learning algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 348–353, Oct. 2008.

[44] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York: Cambridge University Press, 2009.

[45] J. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[46] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and L. A.J., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences of the USA*, vol. 96, no. 12, pp. 6745–6750, 1999.

[47] S. Yu, T. Falck, A. Daemen, L. C. Tranchevent, J. A. K. Suykens, B. D. Moor, and Y. Moreau, "L_2-norm multiple kernel learning and its application to biomedical data fusion," *BMC Bioinformatics*, vol. 11, no. 1, pp. 309–324, 2010.

[48] T. Diethe, D. R. Hardoon, and J. Shawe-Taylor, "Constructing nonlinear discriminants from multiple data views," in *Proc. ECML/PKDD*, Barcelona, Spain, Sep. 2010, pp. 328–343.

[49] S. Shekhar, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Joint sparse representation for robust multimodal biometrics recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, pp. 113–126, Jan. 2014.

Wanqi Yang is a PhD candidate in the Department of Computer Science and Technology, Nanjing University, China. Her research interests include multi-view learning, feature selection, multi-modal fusion, abnormal event detection and activity recognition. She has published several papers in top conferences and journals, e.g., CVIU. Her work currently focuses on multi-view feature selection, cross-view correlation analysis and their applications to real-world problems in image/video analysis.

Yang Gao (M'05) received a Ph.D. degree in computer science and technology from the Department of Computer Science and Technology, Nanjing University, Nanjing, China, in 2000.

He is a Professor with the Department of Computer Science and Technology, Nanjing University. He has published more than 100 papers in top conferences and journals. His current research interests include artificial intelligence and machine learning.

Yinghuan Shi is currently an assistant researcher in the Department of Computer Science and Technology of Nanjing University, China. He received his Ph.D. and B.Sc. degrees from the Department of Computer Science of Nanjing University, in 2013 and 2007, respectively. His research interests include computer vision and biomedical image analysis. He has published 10+ research papers in related journals and conferences such as IEEE Trans. Biomedical Engineering, CVPR and IPMI. He serves as a program committee member for several international conferences.

Longbing Cao is a Professor at the University of Technology Sydney, the founding director of the university's research institute Advanced Analytics Institute, and the Data Mining Research Leader of the Australian Capital Markets Cooperative Research Centre. He received one PhD in Intelligent Sciences and another in Computing Sciences. His research interests include data sciences, big data analytics, machine learning, behavior informatics, multi-agent technology, open complex intelligent systems, agent mining, and their enterprise applications. He is a senior member of IEEE Computer, SMC and CIS societies.