2D Feature Selection by Sparse Matrix Regression

Chenping Hou, *Member, IEEE*, Yuanyuan Jiao, Feiping Nie, Tingjin Luo, and Zhi-Hua Zhou, *Fellow, IEEE*

***Abstract*— For many image processing and computer vision problems, data points are in matrix form. Traditional methods often convert a matrix into a vector and then use vector-based approaches. They will ignore the location of matrix elements and the converted vector often has high dimensionality. How to select features for 2D matrix data directly is still an uninvestigated important issue. In this paper, we propose an algorithm named sparse matrix regression (SMR) for direct feature selection on matrix data. It employs the matrix regression model to accept matrix as input and bridges each matrix to its label. Based on the intrinsic property of regression coefficients, we design some sparse constraints on the coefficients to perform feature selection. An effective optimization method with provable convergence behavior is also proposed. We reveal that the number of regression vectors can be regarded as a tradeoff parameter to balance the capacity of learning and generalization in essence. To examine the effectiveness of SMR, we have compared it with several vector-based approaches on some benchmark data sets. Furthermore, we have also evaluated SMR in the application of scene classification. They all validate the effectiveness of our method.**

***Index Terms*— Two dimensional data, feature selection, sparse matrix regression, scene classification.**

I. Introduction

MATRIX data, or more commonly, tensor data, has emerged in many real applications, especially in the fields of image processing and video analysis. For example, in Scene Classification (SC) [1], there are many images collected from different scenes. The raw pixel-wise representation of each image is a matrix. Thus, the natural representation of each data set is a set of matrices.

Although matrix data arises in many fields, the original matrices are always scanned into vectors in traditional researches, since most of traditional approaches are vector-based [2]. It has been acknowledged that this kind of vectorization will cause some problems. The vector representation of original matrix often has high dimensionality. For example, when the sample images have a little higher resolution, e.g., 128×128, the dimensionality of this reshaped raw pixel-wise vector is 16384. In this scenario, the performances of traditional vector-based methods will degrade [3]–[5]. One reason may be that traditional vector based methods often suffer from small sample size problem [2].

Fig. 1. Sample images from the ORL data set with noisy occlusion.

To handle the high dimensionality problem, dimensionality reduction approaches have been widely investigated in recent years [6]. It aims to reduce the dimensionality of the high dimensional data by finding a set of relevant features. It formulates a smaller set of representative features and retains the optimal salient characteristics. Preprocessing data in this way not only decreases the processing time but also leads to more compactness and better generalization of the learned models [7], [8]. Nevertheless, traditional dimensionality reduction approaches are often vector-based. Compared with the matrix representation, vectorization will ignore the location information of original matrix element [2], [9]–[12]. After vectorization, each element is treated equally in the following tasks. For example, as seen from image in Fig. 1, if we add the block-wise noisy occlusion to the sample images from the ORL data set,[1] we should treat this occlusion in a whole part. This correlation will be lost when they are treated as flatting vectors. Besides, the influence and effects of this noisy occlusion for vector-based approaches, will depend on how we vectorize the image. Thus, it is necessary to investigate the problem of dimensionality reduction for matrix data directly.

Similar to vector based dimensionality reduction approaches, there should be two distinct ways for matrix based dimensionality reduction, i.e., feature extraction [6], [8] and feature selection [7], [13], [14]. Feature extraction combines several original features to form new representations. There are a lot of prominent vector based feature extraction approaches, such as Principal Components Analysis (PCA) [3] and Linear Discriminant Analysis (LDA) [3]. Correspondingly, a lot of efforts have also been devoted to extending these vector-based approaches to manipulate matrix data directly. For example, 2DPCA [15], [16] and $(2D)^2$PCA [17] are the matrix counterparts of PCA, 2DLDA [11] and $(2D)^2$LDA [18]

Manuscript received March 10, 2016; revised February 26, 2017 and May 3, 2017; accepted June 5, 2017. Date of publication June 8, 2017; date of current version July 6, 2017. This work was supported by NSF China under Grant 61333014, Grant 61473302, and Grant 61503396. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vishal Monga. *(Corresponding author: Chenping Hou.)*

C. Hou and T. Luo are with the College of Science, National University of Defense Technology, Changsha 410073, China (e-mail: hcpnudt@hotmail.com; luotingjin_nudt@hotmail.com).

Y. Jiao is with the College of Nine, National University of Defense Technology, Changsha 410073, China (e-mail: jyynudt@gmail.com).

F. Nie is with the Center for OPTical IMagery Analysis and Learning, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: feipingnie@gmail.com).

Z.-H. Zhou is with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: zhouzh@lamda.nju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2713948

[1] http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

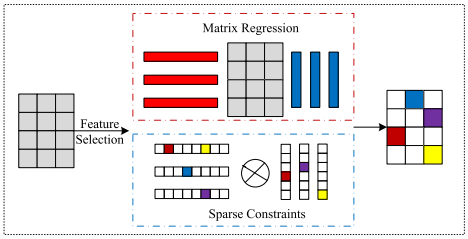


Fig. 2. The intuition of SMR.

are the matrix counterparts of LDA. Besides, many feature extraction algorithms [19]–[23] based on the tensor subspace method have also been developed for data representation, pattern classification and network abnormal detection [24]. For more details, please refer to [25] for a comprehensive review.

Different from feature extraction, feature selection focuses on selecting a few relevant features to represent the original data. It does not change the original representations and maintains the physical meaning of data variables. Consequently, many efforts have been devoted to addressing the problem of feature selection during the past few years [7], [13], [14], [26].

There are plenty of vector-based supervised feature selection algorithms in the literature. We would like to introduce some representative approaches briefly and see more details in Section II. Fisher Score [3] is a widely applied filter-type feature selection algorithm that is based on LDA. T-test [7] is a statistical algorithm used to determine if the distributions of values of a feature for two different classes are distinct. Information Gain (IG) [27] is another popular feature selection approach based on information theory. Another information based method named Maximum Relevance Minimum Redundancy (MRMR) [28] focuses on relevance-redundancy analysis. Besides, ReliefF is another well-known feature weighting algorithm proposed by Kira and Rendell [29]. Recently, Nie *et al.* [30] proposed a Robust Feature Selection algorithm, named as RFS, by incorporating sparse constraint in robust linear regression. Liu *et al.* [13] have proposed a global and local structure preservation feature selection method, name as GLSPFS. Tao *et al.* [26] have enabled traditional feature extraction approach, i.e., LDA, conducting feature selection by adding sparse constraint. For a comprehensive review of supervised feature selection, please refer to the toolbox Weka [31] and Scikit-Feature [32] for more details.

Although a lot of promising algorithms are proposed for feature selection, they are all vector based. As what we have mentioned before, they often ignore the location information, e.g., the block-wise occlusion structure. Each matrix is reshaped as a vector and feature selection algorithm is then employed. It will undoubtedly suffer from the above mentioned problem.

In this paper, we propose a contribution to solve the problem of direct matrix data feature selection by introducing a novel supervised and embedding based feature selection algorithm: Sparse Matrix Regression (SMR). As shown in Fig. 2, to use location information by treating each element unequally, we propose to use matrix regression framework to connect the matrix and its label. For feature selection, by revealing the intrinsic property of matrix regression coefficients, we plan to add some sparse constraints on them to perform feature selection. We provide an effective method to solve the proposed problem with sparse constraints, together with the convergence behavior analysis. Compared with traditional vector-based feature selection approaches, our method has been demonstrated to have better performances on some benchmark data sets. Compared with representative feature extraction approaches, our method can maintain the physical meaning of data variables and achieves comparable or even better performance. Further, we have also evaluated our method on a real scenario, i.e., scene classification. They all validate the effectiveness of SMR. Besides, our algorithm takes matrix data as a demonstration. It can be extended for manipulating any kinds of high order tensor data directly.

The rest of this paper is organized as follows. Section II briefly describes some related works. We formulate the SMR algorithm and provide the corresponding optimization procedure in Section III. The performance analysis, including convergence behaviour and parameter determination, are presented in Section IV. Section V provides some promising comparative results on various kinds of data sets. We evaluate our algorithm on a real task, i.e., scene classification, in Section VI, followed by the conclusions and future works in Section VII.

II. Related Work

In this section, we will briefly review several representative vector-based feature selection approaches. First, let us introduce several notations.

A. Notations

In this paper, matrices and vectors are written in boldface. For a matrix $\mathbf{M} = (m_{ij})$, its i-th row and j-th column are denoted as $\mathbf{m}^i$ and $\mathbf{m}_j$ respectively. Denote $\{\mathbf{X}_i \in \mathbb{R}^{m\times n} | i = 1, 2, \cdots, l\}$ as the set of training examples and the associated class label vectors are $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_l\} \subset \mathbb{R}^c$. Here, $\mathbf{y}_i = [y_{i1}, y_{i2}, \cdots, y_{ic}]^T$. $y_{ij} = 1$ if and only if $\mathbf{X}_i$ belongs to the j-th category and $y_{ij} = 0$ otherwise. c is the number of classes. m and n are the first and second dimensions of each matrix data. l is the number of training points. Define $\mathbf{e} = [1, 1, \cdots, 1] \in \mathbb{R}^{1\times l}$ as a row vector of all ones and $\alpha > 0$ as a balance parameter. Define $\mathrm{Vec}(\cdot)$ as an operator which can convert a matrix to a vector by collecting the columns. Denote $\{\mathbf{x}_i = \mathrm{Vec}(\mathbf{X}_i) | i = 1, 2, \cdots, l\}$ as the vector counterparts of the training examples.

Denote the $\ell_{r,p}$-norm of a matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$ as [30]

$$\|\mathbf{M}\|_{r,p} = \left(\sum_{i=1}^{m} \left(\sum_{j=1}^{n} |m_{ij}|^r \right)^{\frac{p}{r}} \right)^{\frac{1}{p}}, \quad r > 0, p > 0. \quad (1)$$

When $r \geq 1$ and $p \geq 1$, $\ell_{r,p}$-norm is a valid norm as it satisfies the three norm conditions.

B. Representative Supervised Feature Selection Approaches

In the following parts, we will introduce three representative supervised feature selection approaches, i.e., Fisher Score,

MRMR and Robust Feature Selection, which only accept vectors as the input .

1) Fisher Score (FisherScor) [3] is derived from one of the well-known feature extraction algorithms, i.e., LDA. It tries to search a subset of features, such that in the data space spanned by the selected features, the distances between data points in different classes are as large as possible, while the distances between data points in the same class are as small as possible. Concretely, the objective function of Fisher Score is

$$\mathcal{L} = \mathrm{Tr}(\mathbf{S}_b\ (\mathbf{S}_t + \gamma\,\mathbf{I})^{-1}), \tag{2}$$

where γ is a positive regularization parameter. $\mathbf{S}_b$ is called between-class scatter matrix and $\mathbf{S}_t$ is called total scatter matrix. They are calculated based on $\mathbf{z}$, the low dimensional representation of $\mathbf{x}$, using the same formula as in traditional LDA. It is a relevance-only criterion and it does not apply any redundancy analysis on the existing features.

2) Maximum Relevance Minimum Redundancy (MRMR) [28] is a popular relevance-redundancy approach. It selects features that are distant in terms of mutual information and have high correlation to the classification variable according to the minimal-redundancy-maximal-relevance criterion based on mutual information.

3) Robust Feature Selection (RFS) [30] is a novel vector-based feature selection approach. It emphasizes sparse constraints on both loss function and regularization. The proposed method is robust to outliers in data points and the sparse regularization selects features across all data points. Concretely, the objective function of RFS is

$$\mathcal{L}(\mathbf{W}, \mathbf{b}) = \sum_{i=1}^{l} \|\mathbf{W}^T\mathbf{x}_i + \mathbf{b} - \mathbf{y}_i\|_F + \alpha\|\mathbf{W}\|_{2,1}. \tag{3}$$

After deriving the optimal solution, RFS uses the 2-norm of the row vectors of $\mathbf{W}$ to measure the importance of each feature and to select the most important features. Although RFS achieves prominent performances in many applications, it only accepts vectors as its input.

C. Representative Regression Method for Matrix Data

General Bilinear Regression (GBR) [33] is a regression model which takes matrix data as the input. It is the two-dimensional counterpart of traditional vector based regression method. It replaces the regression function of traditional model by a bilinear regression function. More concretely, in a two-class scenario, we assume that the left and right projection vectors are $\mathbf{u}$ and $\mathbf{v}$. Its objective function is

$$\mathcal{L}(\mathbf{u}, \mathbf{v}, b) = \textstyle\sum_{i=1}^{l} \|\mathbf{u}^T\mathbf{X}_i\mathbf{v} + b - y_i\|^2.$$

GBR has only been analysed mathematically. Besides, it only uses one left projecting vector together with one right projecting vector. Its fitting error is too large for some real regression problems.

III. Sparse Matrix Regression

Assume that we have been given l training examples, denoted as $\{\mathbf{X}_i \in \mathbb{R}^{m\times n} | i = 1, 2, \cdots, l\}$. The associated class label vectors are $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_l\} \subset \mathbb{R}^c$. We want to use the label information in selecting important features from these matrix data directly.

A. Formulation

Considering that the input of our algorithm is matrix and the label information is also available, we try to use both of them by employing regression model since the regression methods perform well for the task of classifier training.

The objective function of SMR is composed of two parts. Generally, the first one measures the loss with matrix regression and the other is the sparse constraint designed for feature selection. Before going into the details, we reformulate the objective function of traditional regularized least square regression method as follows,

$$\begin{aligned}\mathcal{L}(\mathbf{W}, \mathbf{b}) &= \sum_{i=1}^{l} \|\mathbf{W}^T\mathbf{x}_i + \mathbf{b} - \mathbf{y}_i\|_F^2 + \alpha\|\mathbf{W}\|_F^2 \\ &= \sum_{i=1}^{l}\sum_{r=1}^{c} (\mathbf{w}_r^T\mathbf{x}_i + b_r - y_{ir})^2 + \alpha\sum_{r=1}^{c}(\mathbf{w}_r^T\mathbf{w}_r) \\ &= \sum_{r=1}^{c}\left(\sum_{i=1}^{l} (\mathbf{w}_r^T\mathbf{x}_i + b_r - y_{ir})^2 + \alpha(\mathbf{w}_r^T\mathbf{w}_r)\right),\end{aligned} \tag{4}$$

where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_c] \in \mathbb{R}^{mn\times c}$ and $\mathbf{b} = [b_1, b_2, \cdots, b_c]^T$ are the projection matrix and constant bias respectively, $\mathbf{x}_i = \mathrm{vec}(\mathbf{X}_i)$ is the vectorization of $\mathbf{X}_i$.

As it can be seen from the above deduction, we can separate the original regularized least regression model into c subproblems. In other words, the formulation in (4) can be regarded as training c classifiers for c categories separately. Thus, in the following, we only consider to train a classifier for the r-th category.

When the inputs are changed to matrices, one direct way for extending the loss function of traditional regression model shown in (4) is to replace the traditional projection term, i.e., $\mathbf{w}_r^T\mathbf{x}_i$ in (4), by its tensor counterpart, i.e., $\mathbf{u}_r^T\mathbf{X}_i\mathbf{v}_r$, where $\mathbf{u}_r$ and $\mathbf{v}_r$ are the left and right transformation vectors for the r-th category. This is called General Bilinear Regression in the literature [33]. Nevertheless, this kind of replacement will induce some strong constraints since

$$\begin{aligned}\mathbf{u}_r^T\mathbf{X}_i\mathbf{v}_r &= \mathrm{Tr}(\mathbf{u}_r^T\mathbf{X}_i\mathbf{v}_r) = \mathrm{Tr}(\mathbf{X}_i\mathbf{v}_r\mathbf{u}_r^T) \\ &= \mathrm{Tr}((\mathbf{u}_r\mathbf{v}_r^T)^T\mathbf{X}_i) = \mathrm{Vec}(\mathbf{u}_r\mathbf{v}_r^T)^T\mathrm{Vec}(\mathbf{X}_i).\end{aligned} \tag{5}$$

Comparing (4) with the GBR formulation, we know that the vector counterpart of $\mathbf{w}_r$ is $\mathrm{Vec}(\mathbf{u}_r\mathbf{v}_r^T)$. In other words, the mn values in a tensor base $\mathbf{u}_r\mathbf{v}_r^T$ of GBR only have $m+n$ degrees of free variables. In many real cases, these added constraints are too strict. It can not characterize the original data fully and thus, increases the regression error.

To solve this problem, we discuss it from the view of learning theory at first. The GBR model has large training errors and thus, its learning capacity is poor. One direct way for enhancing the learning capacity is increasing the model complexity. In other words, we want to increase the number of free variables and relax these too strict constraints. Instead of

using merely one couple of projecting vectors, i.e., the left projecting vector $\mathbf{u}_r$ and right projecting vector $\mathbf{v}_r$ for the r-th classifier, we propose to use k couples of left projecting vectors and right projecting vectors as in our previous work [34]. They are denoted as $\{\mathbf{u}_j^{(r)}\}_{j=1}^k$ and $\{\mathbf{v}_j^{(r)}\}_{j=1}^k$. Here, k is the number of regression vectors and it is the parameter to balance the capacity of learning and generalization for the regression model in (6). Intuitively, when k is large, we have more independent optimization variables. It means that the model has small training error and strong learning capacity. Nevertheless, the model trends to be over-fitting and its generalization capacity is weak. We will reveal the essence of k in Proposition 2. See more details in Section IV (B).

Besides, we can also discuss it from the view of ensemble learning. We use k couples of regression vectors and join them in formulating the regression item. Essentially, it can be regarded as using an ensemble strategy to reduce the variance of classifier predictions on different types of data sets [35].

In particular, the loss function for the r-th classifier is

$$\sum_{i=1}^{l}\left(\sum_{j=1}^{k}(\mathbf{u}_j^{(r)})^T\mathbf{X}_i\mathbf{v}_j^{(r)}+b_r-y_{ir}\right)^2, \tag{6}$$

where b_r is the unknown bias for the r-th category.

Denote $\mathbf{U}^{(r)}=[\mathbf{u}_1^{(r)},\mathbf{u}_2^{(r)},\cdots,\mathbf{u}_k^{(r)}]\in\mathbb{R}^{m\times k}$ and $\mathbf{V}^{(r)}=[\mathbf{v}_1^{(r)},\mathbf{v}_2^{(r)},\cdots,\mathbf{v}_k^{(r)}]\in\mathbb{R}^{n\times k}$, the loss function in (6) can be reformulated as

$$\sum_{i=1}^{l}\left(\mathrm{Tr}((\mathbf{U}^{(r)})^T\mathbf{X}_i\mathbf{V}^{(r)})+b_r-y_{ir}\right)^2. \tag{7}$$

The second objective function is designed for feature selection. Inspired by the basic idea of sparse regression for feature selection in [30], we also want to add sparse constraints on the transformation matrix to measure their values in regression. Unfortunately, in matrix regression, we have two groups of regression vectors and their relationships are close. It is unwise to add sparse constraints on them separately. In the following, we will design a new sparse constraint on their combinations.

Note that, for the r-th classifier, we have

$$\begin{aligned}\sum_{j=1}^{k}(\mathbf{u}_j^{(r)})^T\mathbf{X}_i\mathbf{v}_j^{(r)}&=\sum_{j=1}^{k}\mathrm{Tr}((\mathbf{u}_j^{(r)})^T\mathbf{X}_i\mathbf{v}_j^{(r)})\\&=\sum_{j=1}^{k}\mathrm{Tr}(\mathbf{X}_i\mathbf{v}_j^{(r)}(\mathbf{u}_j^{(r)})^T)=\mathrm{Tr}(\mathbf{X}_i\ (\sum_{j=1}^{k}\mathbf{v}_j^{(r)}(\mathbf{u}_j^{(r)})^T))\\&=\mathrm{Tr}(\mathbf{X}_i\mathbf{V}^{(r)}(\mathbf{U}^{(r)})^T)=\mathrm{Tr}((\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T)^T\mathbf{X}_i)\\&=(\mathrm{Vec}(\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T))^T\mathrm{Vec}(\mathbf{X}_i).\end{aligned} \tag{8}$$

Similar to the observations from (5), in matrix regression model, the tensor counterpart of $\mathbf{w}_r$ in (4) is $\mathrm{Vec}(\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T)$. Denote $\mathbf{p}_r=\mathrm{Vec}(\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T)$ and $\mathbf{P}=[\mathbf{p}_1,\mathbf{p}_2,\cdots,\mathbf{p}_c]$. Then, the tensor counterpart of $\mathbf{W}$ in (4) should be $\mathbf{P}\in\mathbb{R}^{mn\times c}$. It can be regarded as the linear transformation matrix as in traditional regression model. The r-th column vector $\mathbf{p}_r$ corresponds to the classifier trained by one versus rest strategy. In other aspect, we can also regard $\mathbf{P}$ as the transformation matrix which can also perform dimensionality reduction. The i-th row vector, denoted by $\hat{\mathbf{p}}_i$, corresponds to the transformation vector of the i-th feature in regression for $i=1,2,\cdots,mn$. It can be regarded as a vector that measures the importance of the i-th feature. For convenience, let

$$\mathbf{P}=\left[\hat{\mathbf{p}}_1^T,\hat{\mathbf{p}}_2^T,\cdots,\hat{\mathbf{p}}_{mn}^T\right]^T. \tag{9}$$

Since our task is feature selection, we expect that the transformation matrix $\mathbf{P}$ holds some structure sparsity property. Concretely, we expect that most $\hat{\mathbf{p}}_i$ are zeros. In detail, the corresponding features can be neglected since these features are redundant for regression. When we use the 2-norm of $\hat{\mathbf{p}}_i$ as a metric to measure its contribution in regression, the sparsity property, i.e., a small number of $\hat{\mathbf{p}}_i$ entries are non-zero, indicates the following objective function,

$$\arg\min_{\mathbf{P}}\sum_{i=1}^{mn}(\|\hat{\mathbf{p}}_i\|_2)^p=\sum_{i=1}^{mn}\left(\sum_{j=1}^{c}\left|p_{ij}\right|^2\right)^{p/2}=\|\mathbf{P}\|_{2,p}^p, \tag{10}$$

with $0<p\le 1$ for the sake of feature selection. It is a sparse constraint and requires that a small number of $\hat{\mathbf{p}}_i$ are non-zeros vectors. The non-zero $\hat{\mathbf{p}}_i$ corresponds to the important features since the $\hat{\mathbf{p}}_i$ with all zero elements can be neglected in the former regression.

By combining the objective functions in (7), (10) and joining all categories, the Sparse Matrix Regression (SMR) algorithm for feature selection can be summarized as follows.

$$\begin{aligned}&\mathcal{L}(\mathbf{U}^{(1)},\cdots,\mathbf{U}^{(c)},\mathbf{V}^{(1)},\cdots,\mathbf{V}^{(c)},\mathbf{b})\\&=\sum_{r=1}^{c}\sum_{i=1}^{l}\left(\mathrm{Tr}((\mathbf{U}^{(r)})^T\mathbf{X}_i\mathbf{V}^{(r)})+b_r-y_{ir}\right)^2+\alpha\|\mathbf{P}\|_{2,p}^p\\&\quad\text{with }\mathbf{P}=[\mathbf{p}_1,\mathbf{p}_2,\cdots,\mathbf{p}_c],\ \mathbf{p}_r=\mathrm{Vec}(\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T),\end{aligned} \tag{11}$$

where α is a non-negative balance parameter. It can be determined by traditional parameter determination approaches, such as cross validation. In our experimental results, we predefine a parameter set and search the best α from this set. See more details in the end of Section V(A).

Comparing the SMR formulation in (11) with the traditional regression model in (4), we would like to explain why SMR could use location information of matrix data. Intuitively, each row (or column) vector is treated equally in (7). The block-wise occlusion in the images shown in Fig. 1 is treated as a whole part. Besides, as seen from (8), if we use traditional regression model as in (4), it is obvious that all elements in $\mathrm{Vec}(\mathbf{X}_i)$ are treated equally. In SMR, however, the counterpart of traditional regression coefficient is $\mathrm{Vec}(\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T)$ and different elements of $\mathrm{Vec}(\mathbf{X}_i)$ correspond to a different regression coefficient constraint. In other words, if we permutate the elements of the original matrix in different ways, the outputs will be different. Thus, SMR could use location information to some extent.

After deriving the optimal solution, we use the 2-norm of $\hat{\mathbf{p}}_i$, i.e., $\|\hat{\mathbf{p}}_i\|_2$, to evaluate the importance of each feature. Each feature corresponds to an element of the matrix. Thus, the larger this value is, the more important this element is.

In real-world applications, we can either select a fixed number of the most important elements or set a threshold and select the element whose importance is larger than this value. In the following, we select a fixed number, i.e., s, features for evaluation.

B. Solution

As seen from the optimization problem in (11), it is difficult to solve this problem directly since 1) both of the terms are non-smooth; 2) the sparse constraints are added to $\mathbf{P}$, whose elements are the complex combinations of $\mathbf{U}^{(r)}$ and $\mathbf{V}^{(r)}$ for $r = 1, 2, \cdots, c$. Besides, two groups of optimization variables, i.e., $\{\mathbf{U}^{(r)}\}_{r=1}^{c}$ and $\{\mathbf{V}^{(r)}\}_{r=1}^{c}$ are coupled with each other in two ways. First, $\mathbf{U}^{(r)}$ and $\mathbf{V}^{(r)}$ are matrix regression matrices, and they are coupled in the loss function. Second, all the $\mathbf{U}^{(r)}$ and $\mathbf{V}^{(r)}$ are coupled together in formulating the regularizer $\mathbf{P}$. Thus, it is difficult to solve them simultaneously and we propose to optimize them alternatively. Our theoretical results show that this kind of iteration will converge.

1) Fix $\{\mathbf{V}^{(r)}\}_{r=1}^{c}$ and Optimize $\{\mathbf{U}^{(r)}\}_{r=1}^{c}$ and $\{b_r\}_{r=1}^{c}$: When all $\mathbf{V}^{(r)}$ are fixed, we need to optimize a set of $\mathbf{U}^{(r)}$ and each $\mathbf{U}^{(r)}$ contains multiple regression vectors. The optimization problem seems complicated. Nevertheless, after some deductions, we can reformulate the problem in (11) and derive its solution effectively.

Recalling the basic idea to solve the sparse constraints problem as in [30], we take the derivative of $\|\mathbf{P}\|_{2,p}^{p}$ with respect to $\mathbf{P}$. For convenience, we denote $\mathcal{L}(\mathbf{P}) = \|\mathbf{P}\|_{2,p}^{p}$. When $\hat{\mathbf{p}}_i \neq \mathbf{0}$ for $i = 1, 2, \cdots, mn$, the derivative of $\mathcal{L}(\mathbf{P})$ with respect to $\mathbf{P}$ is

$$\frac{\partial \mathcal{L}(\mathbf{P})}{\partial \mathbf{P}} = 2\mathbf{D}^{(v)}\mathbf{P}, \tag{12}$$

where $\mathbf{D}^{(v)} \in \mathbb{R}^{mn \times mn}$ is a diagonal matrix with the i-th diagonal element as

$$d_{ii}^{(v)} = \frac{p}{2}\left\|\hat{\mathbf{p}}_i\right\|_2^{p-2}. \tag{13}$$

Here $\hat{\mathbf{p}}_i$ is the i-th row of $\mathbf{P}$ as defined in (9).

When $\mathbf{D}^{(v)}$ is fixed, the derivative of $\mathcal{L}$ in (11) can also be regarded as the derivative of the following objective function.

$$\begin{aligned}&\mathcal{L}(\mathbf{U}^{(1)}, \cdots, \mathbf{U}^{(c)}, \mathbf{V}^{(1)}, \cdots, \mathbf{V}^{(c)}, \mathbf{b})\\ &= \sum_{r=1}^{c}\sum_{i=1}^{l}\left(\mathrm{Tr}((\mathbf{U}^{(r)})^T\mathbf{X}_i\mathbf{V}^{(r)}) + b_r - y_{ir}\right)^2 + \alpha\mathrm{Tr}(\mathbf{P}^T\mathbf{D}^{(v)}\mathbf{P})\\ &\quad \text{with } \mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_c],\ \mathbf{p}_r = \mathrm{Vec}(\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T).\end{aligned} \tag{14}$$

In the following, we use the objective function in (14) to approximate the SMR formulation in (11). By solving this problem with our method, we will prove that the objective function of SMR will also decrease. Moreover, with this approximation, we can solve the problem in (14) effectively.

Before going into the details for solving the problem in (14), we decompose it into c separate sub-problems. After some deductions, the following equation holds,

$$\mathrm{Tr}(\mathbf{P}^T\mathbf{D}^{(v)}\mathbf{P}) = \sum_{r=1}^{c}\mathbf{p}_r^T\mathbf{D}^{(v)}\mathbf{p}_r. \tag{15}$$

Thus, when $\mathbf{D}^{(v)}$ is fixed, we can decompose the objective function in (14) into the following c independent sub-problems

$$\begin{aligned}&\arg\min \mathcal{L}(\mathbf{U}^{(r)}, \mathbf{V}^{(r)}, b_r)\\ &= \sum_{i=1}^{l}\left(\mathrm{Tr}((\mathbf{U}^{(r)})^T\mathbf{X}_i\mathbf{V}^{(r)}) + b_r - y_{ir}\right)^2 + \alpha\mathbf{p}_r^T\mathbf{D}^{(v)}\mathbf{p}_r.\end{aligned} \tag{16}$$

Denote

$$\mathbf{f}_i^{(r)} = \begin{bmatrix}\mathbf{X}_i\mathbf{v}_1^{(r)}\\ \mathbf{X}_i\mathbf{v}_2^{(r)}\\ \vdots\\ \mathbf{X}_i\mathbf{v}_k^{(r)}\end{bmatrix}_{mk\times 1}, \quad \hat{\mathbf{u}}^{(r)} = \begin{bmatrix}\mathbf{u}_1^{(r)}\\ \mathbf{u}_2^{(r)}\\ \vdots\\ \mathbf{u}_k^{(r)}\end{bmatrix}_{mk\times 1}. \tag{17}$$

Then, the loss function in (16) is equivalent to

$$\sum_{i=1}^{l}\left((\hat{\mathbf{u}}^{(r)})^T\mathbf{f}_i^{(r)} + b_r - y_{ir}\right)^2. \tag{18}$$

Note that $\hat{\mathbf{u}}^{(r)} = \mathrm{Vec}(\mathbf{U}^{(r)})$, then $\mathbf{p}_r$ can be reformulated with respect to $\hat{\mathbf{u}}^{(r)}$ as follows.

$$\begin{aligned}\mathbf{p}_r &= \mathrm{Vec}(\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T) = \mathrm{Vec}(\mathbf{I}\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T)\\ &= (\mathbf{V}^{(r)}\otimes\mathbf{I})\mathrm{Vec}(\mathbf{U}^{(r)}) = (\mathbf{V}^{(r)}\otimes\mathbf{I})\hat{\mathbf{u}}^{(r)},\end{aligned} \tag{19}$$

where $\otimes$ denotes the Kronecker product and $\mathbf{I}$ represents the identity matrix.

The regularization term in (16) can be reformulated as

$$\begin{aligned}\mathbf{p}_r^T\mathbf{D}^{(v)}\mathbf{p}_r &= (\hat{\mathbf{u}}^{(r)})^T(\mathbf{V}^{(r)}\otimes\mathbf{I})^T\mathbf{D}^{(v)}(\mathbf{V}^{(r)}\otimes\mathbf{I})\hat{\mathbf{u}}^{(r)}\\ &= (\hat{\mathbf{u}}^{(r)})^T\mathbf{A}^{(r)}\hat{\mathbf{u}}^{(r)},\end{aligned} \tag{20}$$

where $\mathbf{A}^{(r)} = (\mathbf{V}^{(r)}\otimes\mathbf{I})^T\mathbf{D}^{(v)}(\mathbf{V}^{(r)}\otimes\mathbf{I})$.

Denote

$$\begin{aligned}\mathbf{F}^{(r)} &= [\mathbf{f}_1^{(r)}, \mathbf{f}_2^{(r)}, \cdots, \mathbf{f}_l^{(r)}]_{mk\times l},\\ \mathbf{y}^{(r)} &= [y_{1r}, y_{2r}, \cdots, y_{lr}]_{1\times l}.\end{aligned} \tag{21}$$

Then, (16) becomes

$$\begin{aligned}&\arg\min \mathcal{L}(\hat{\mathbf{u}}^{(r)}, b_r)\\ &= ((\hat{\mathbf{u}}^{(r)})^T\mathbf{F}^{(r)} + b_r\mathbf{e} - \mathbf{y}^{(r)})((\hat{\mathbf{u}}^{(r)})^T\mathbf{F}^{(r)} + b_r\mathbf{e} - \mathbf{y}^{(r)})^T\\ &\quad + \alpha\ (\hat{\mathbf{u}}^{(r)})^T\mathbf{A}^{(r)}\hat{\mathbf{u}}^{(r)}.\end{aligned} \tag{22}$$

where $\mathbf{e}$ is a vector whose elements are all 1.

It is a regularized least squares problem. Take the derivative of $\mathcal{L}(\hat{\mathbf{u}}^{(r)}, b_r)$ with respect to $\hat{\mathbf{u}}^{(r)}$, b_r and set it to zero. The optimal solutions for (16) are

$$\begin{aligned}\hat{\mathbf{u}}^{(r)} &= \left[\mathbf{F}^{(r)}\mathbf{L}_c\ (\mathbf{F}^{(r)})^T + \alpha\mathbf{A}^{(r)}\right]^{-1}\mathbf{F}^{(r)}\mathbf{L}_c\ (\mathbf{F}^{(r)})^T,\\ b_r &= \frac{1}{l}\left(\mathbf{y}^{(r)} - (\hat{\mathbf{u}}^{(r)})^T\mathbf{F}^{(r)}\right)\mathbf{e}^T,\end{aligned} \tag{23}$$

where $\mathbf{L}_c = \mathbf{I} - \frac{1}{l}\mathbf{e}^T\mathbf{e}$.

In words, when $\{\mathbf{v}_i^{(r)}\}_{i=1}^{k}$ is fixed, we have approximated the optimization problem in (11) as the problem in (14), which can be solved in an alternative way. Concretely, we alternatively update $\mathbf{D}^{(v)}$ using (13) and update $\hat{\mathbf{u}}^{(r)}$, b_r using (23). The solutions calculated by (23) are also the global optimization to (14), provided that $\mathbf{D}^{(v)}$ is fixed.

2) Fix $\{\mathbf{U}^{(r)}\}_{r=1}^{c}$ and Optimize $\{\mathbf{V}^{(r)}\}_{r=1}^{c}$ and $\{b_r\}_{r=1}^{c}$: As seen from the formulation of SMR in (11), we also want to reformulate it as a regularized least squares problem as shown in (22). Nevertheless, the vector $\mathbf{p}_r$ in (19) can only be formulated by $\mathrm{Vec}((\mathbf{V}^{(r)})^T)$, not $\mathrm{Vec}(\mathbf{V}^{(r)})$, as follows.

$$\begin{aligned}\mathbf{p}_r &= \mathrm{Vec}(\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T) = \mathrm{Vec}(\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T\mathbf{I})\\ &= (\mathbf{I}\otimes\mathbf{U}^{(r)})\mathrm{Vec}((\mathbf{V}^{(r)})^T). \end{aligned} \tag{24}$$

Since $\mathbf{p}_r$ can not be formulated by $\mathrm{Vec}(\mathbf{V}^{(r)})$, the above deduction can not be implemented when $\mathbf{U}^{(r)}$ is fixed. To solve this problem, we go back to the original formulation of SMR in (11). In the formulation of $\mathbf{p}_r$, $\mathbf{U}^{(r)}$ and $\mathbf{V}^{(r)}$ are not changeable. This is the main reason for our difficulty in solving SMR as previous. Fortunately, after some deductions, we have the following equations, which can facilitate our solution.

$$\begin{aligned}\|\mathbf{P}\|_{2,p}^{p} &= \|\mathbf{Q}\|_{2,p}^{p}\\ \mathbf{P} &= [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_c],\\ \mathbf{p}_r &= \mathrm{Vec}(\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T);\\ \mathbf{Q} &= [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_c],\\ \mathbf{q}_r &= \mathrm{Vec}(\mathbf{V}^{(r)}(\mathbf{U}^{(r)})^T). \end{aligned} \tag{25}$$

$$\mathrm{Tr}((\mathbf{U}^{(r)})^T\mathbf{X}_i\mathbf{V}^{(r)}) = \mathrm{Tr}((\mathbf{V}^{(r)})^T\mathbf{X}_i^T\mathbf{U}^{(r)}). \tag{26}$$

(25) holds since $(\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T)^T = \mathbf{V}^{(r)}(\mathbf{U}^{(r)})^T$ and $\mathrm{Vec}(\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T)$, $\mathrm{Vec}(\mathbf{V}^{(r)}(\mathbf{U}^{(r)})^T)$ are different permutations of the same elements (all the elements of $\mathbf{U}^{(r)}(\mathbf{V}^{(r)})^T$ are the same as those of $\mathbf{V}^{(r)}(\mathbf{U}^{(r)})^T$). Since the elements in each column of $\mathbf{P}$ are permutated in the same order as in formulating $\mathbf{Q}$, the rows of $\mathbf{Q}$ are just different arrangements of the rows of $\mathbf{P}$. Besides, the order of rows for a matrix does not change its $\ell_{2,p}$-norm. (26) holds since $\mathrm{Tr}(\mathbf{C}) = \mathrm{Tr}(\mathbf{C}^T)$ for any square matrix $\mathbf{C}$.

Considering the results in (25) and (26), we can reformulate the objective function of SMR in (11) as

$$\begin{aligned}&\mathcal{L}(\mathbf{U}^{(1)}, \cdots, \mathbf{U}^{(c)}, \mathbf{V}^{(1)}, \cdots, \mathbf{V}^{(c)}, \mathbf{b})\\ &= \sum_{r=1}^{c}\sum_{i=1}^{l}\left(\mathrm{Tr}((\mathbf{V}^{(r)})^T\mathbf{X}_i^T\mathbf{U}^{(r)}) + b_r - y_{ir}\right)^2 + \alpha\|\mathbf{Q}\|_{2,p}^{p}\\ &\quad \text{with } \mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_c],\ \mathbf{q}_r = \mathrm{Vec}(\mathbf{V}^{(r)}(\mathbf{U}^{(r)})^T). \end{aligned} \tag{27}$$

Comparing (27) with (11), we now use the same strategies to derive the optimal $\mathbf{V}^{(r)}$ and b_r when $\mathbf{U}^{(r)}$ is fixed. The approximated problem can be derived in the same way as previous. Concretely, the derivative of $\mathcal{L}$ in (27) can also be regarded as the derivative of the following objective function.

$$\begin{aligned}&\mathcal{L}(\mathbf{U}^{(1)}, \cdots, \mathbf{U}^{(c)}, \mathbf{V}^{(1)}, \cdots, \mathbf{V}^{(c)}, \mathbf{b})\\ &= \sum_{r=1}^{c}\sum_{i=1}^{l}\left(\mathrm{Tr}((\mathbf{V}^{(r)})^T\mathbf{X}_i^T\mathbf{U}^{(r)}) + b_r - y_{ir}\right)^2 + \alpha\mathrm{Tr}(\mathbf{Q}^T\mathbf{D}^{(u)}\mathbf{Q})\\ &\quad \text{with } \mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_c],\ \mathbf{q}_r = \mathrm{Vec}(\mathbf{V}^{(r)}(\mathbf{U}^{(r)})^T), \end{aligned} \tag{28}$$

where $\mathbf{D}^{(u)} \in \mathbb{R}^{mn\times mn}$ is a diagonal matrix with the i-th diagonal element as

$$d_{ii}^{(u)} = \frac{p}{2}\left\|\hat{\mathbf{q}}_i\right\|_2^{p-2}. \tag{29}$$

Here, $\hat{\mathbf{q}}_i$ is the i-th row of $\mathbf{Q}$.

The optimization problem in (28) can also be separated into c independent sub-problems as follows,

$$\begin{aligned}&\mathcal{L}(\mathbf{U}^{(r)}, \mathbf{V}^{(r)}, b_r)\\ &= \sum_{i=1}^{l}\left(\mathrm{Tr}((\mathbf{V}^{(r)})^T\mathbf{X}_i^T\mathbf{U}^{(r)}) + b_r - y_{ir}\right)^2 + \alpha\mathbf{q}_r^T\mathbf{D}^{(u)}\mathbf{q}_r. \end{aligned} \tag{30}$$

Denote

$$\begin{aligned}\mathbf{g}_i^{(r)} &= \begin{bmatrix}\mathbf{X}_i^T\mathbf{u}_1^{(r)}\\ \mathbf{X}_i^T\mathbf{u}_2^{(r)}\\ \vdots\\ \mathbf{X}_i^T\mathbf{u}_k^{(r)}\end{bmatrix}_{nk\times 1}, \hat{\mathbf{v}}^{(r)} = \begin{bmatrix}\mathbf{v}_1^{(r)}\\ \mathbf{v}_2^{(r)}\\ \vdots\\ \mathbf{v}_k^{(r)}\end{bmatrix}_{nk\times 1},\\ \mathbf{G}^{(r)} &= [\mathbf{g}_1^{(r)}, \mathbf{g}_2^{(r)}, \cdots, \mathbf{g}_l^{(r)}]_{nk\times l}. \end{aligned} \tag{31}$$

We now reformulated the regularizer in (30) with respect to $\hat{\mathbf{v}}^{(r)} = \mathrm{Vec}(\mathbf{V}^{(r)})$. Note that $\hat{\mathbf{v}}^{(r)} = \mathrm{Vec}(\mathbf{V}^{(r)})$, $\mathbf{q}_r$ can be expressed by $\hat{\mathbf{v}}^{(r)}$ as follows.

$$\begin{aligned}\mathbf{q}_r &= \mathrm{Vec}(\mathbf{V}^{(r)}(\mathbf{U}^{(r)})^T) = \mathrm{Vec}(\mathbf{I}\mathbf{V}^{(r)}(\mathbf{U}^{(r)})^T)\\ &= (\mathbf{U}^{(r)}\otimes\mathbf{I})\mathrm{Vec}(\mathbf{V}^{(r)}) = (\mathbf{U}^{(r)}\otimes\mathbf{I})\hat{\mathbf{v}}^{(r)}. \end{aligned} \tag{32}$$

Then, the regularizer in (30) can be reformulated as

$$\begin{aligned}\mathbf{q}_r^T\mathbf{D}^{(u)}\mathbf{q}_r &= (\hat{\mathbf{v}}^{(r)})^T(\mathbf{U}^{(r)}\otimes\mathbf{I})^T\mathbf{D}^{(u)}(\mathbf{U}^{(r)}\otimes\mathbf{I})\hat{\mathbf{v}}^{(r)}\\ &= (\hat{\mathbf{v}}^{(r)})^T\mathbf{B}^{(r)}\hat{\mathbf{v}}^{(r)}, \end{aligned} \tag{33}$$

where $\mathbf{B}^{(r)} = (\mathbf{U}^{(r)}\otimes\mathbf{I})^T\mathbf{D}^{(u)}(\mathbf{U}^{(r)}\otimes\mathbf{I})$.

With these notations, (30) becomes

$$\begin{aligned}&\arg\min \mathcal{L}(\hat{\mathbf{v}}^{(r)}, b_r)\\ &= ((\hat{\mathbf{v}}^{(r)})^T\mathbf{G}^{(r)} + b_r\mathbf{e} - \mathbf{y}^{(r)})((\hat{\mathbf{v}}^{(r)})^T\mathbf{G}^{(r)} + b_r\mathbf{e} - \mathbf{y}^{(r)})^T\\ &\quad + \alpha\ (\hat{\mathbf{v}}^{(r)})^T\mathbf{B}^{(r)}\hat{\mathbf{v}}^{(r)}. \end{aligned} \tag{34}$$

The optimal solutions should be

$$\begin{aligned}\hat{\mathbf{v}}^{(r)} &= \left[\mathbf{G}^r\mathbf{L}_c\ (\mathbf{G}^{(r)})^T + \alpha\mathbf{B}^{(r)}\right]^{-1}\mathbf{G}^{(r)}\mathbf{L}_c\ (\mathbf{y}^{(r)})^T,\\ b_r &= \frac{1}{l}\left(\mathbf{y}^{(r)} - (\hat{\mathbf{v}}^{(r)})^T\mathbf{G}^{(r)}\right)\mathbf{e}^T, \end{aligned} \tag{35}$$

where $\mathbf{L}_c = \mathbf{I} - \frac{1}{l}\mathbf{e}^T\mathbf{e}$.

In conclusion, when we fix one parameter and optimize with respect to the other, we can derive the solutions in a closed form. Certainly, we can also try to optimize all the parameters simultaneously. Nevertheless, since the optimization parameters are coupled with each other, it is difficult to design the corresponding strategy.

3) SMR Procedure and Highlights: There are several points that should be highlighted here.

1) In the above deduction, although the computed vectors are $\hat{\mathbf{u}}^{(r)}$ and $\hat{\mathbf{v}}^{(r)}$, not $\mathbf{U}^{(r)}$ and $\mathbf{V}^{(r)}$, they are just different arrangements of the same vectors. We can reshape $\hat{\mathbf{u}}^{(r)}$, $\hat{\mathbf{v}}^{(r)}$ into $\mathbf{U}^{(r)}$, $\mathbf{V}^{(r)}$ and use them to formulate $\mathbf{P}$ and $\mathbf{Q}$. Then, we can use the 2-norm of the row vectors to rank the features.

2) In the above procedures, when one group of vectors is fixed, we need not to derive the optimal solutions. In each

round, we only need to update them once. Specifically, we update all the variables using (13), (23), (29) and (35) in sequence and we need not to iterate (13) and (23) to derive the optimal solution to the problem in (14). We will prove that this iteration will also converge.

3) When computing $\mathbf{D}^{(v)}$, its diagonal element $d_{ii}^{(v)}$ is $\frac{p}{2}\left\|\hat{\mathbf{p}}^i\right\|_2^{p-2}$. In practice, $\left\|\hat{\mathbf{p}}^i\right\|_2$ could be very close to zero but not zero. However, $\left\|\hat{\mathbf{p}}^i\right\|_2$ can be zero theoretically. In this case, $d_{ii}^{(v)}=0$ is a sub-gradient of $\|\mathbf{P}\|_{2,p}^p$ w.r.t $\hat{\mathbf{p}}^i$. We can not set $d_{ii}^{(v)}=0$ when $\hat{\mathbf{p}}^i=\mathbf{0}$, otherwise the derived algorithm will not be guaranteed to converge. Instead, we regularize $d_{ii}^{(v)}$ as $d_{ii}^{(v)}=\frac{p}{2}\left((\hat{\mathbf{p}}^i)^T\hat{\mathbf{p}}^i+\zeta\right)^{\frac{p}{2}-1}$. It is easy to see that $\sum_{i=1}^{mn}\left((\hat{\mathbf{p}}^i)^T\hat{\mathbf{p}}^i+\zeta\right)^{\frac{p}{2}}$ approximates $\|\mathbf{P}\|_{2,p}^p$ when $\zeta\to 0$. Similarly, we can use the same strategy in computing $\mathbf{D}^{(u)}$.

4) The fourth one is about initialization. As seen from the above procedure, SMR is solved in an iterative way. We would like to initialize $\mathbf{V}$ by the same way as in [11]. The initialization is $[\mathbf{I}_{k\times k},\mathbf{0}_{k\times\ (n-k)}]^T$. As stated in [11], this kind of initialization performs well. We initialize $\mathbf{D}^{(v)}=\mathbf{I}$ since each feature has the same importance at the beginning. In the following experiments, we use this kind of initialization, unless stated otherwise.

5) As seen from the formulation of SMR, the selected features facilitate the process of matrix regression in essence. It is not selected for the following classification tasks. Thus, when we determine the features, we can also use these features to compute the c-dimensional representation of each point. These representations then can be used for the following procedures, such as classification or clustering. In this case, SMR can also be regarded as a feature extraction algorithm. In the following evaluation, by employing different methods, we use the selected original features as the representation for a fair comparison.

6) In the above iterations, the stopping criterion is that the difference of objective function values between two adjacent iterations is small enough (10^{-6} in our experiments) or the number of iterations is large enough (30 in our experiments). As for the determination of s, it is difficult to determine it without prior. We can specify it according to the real requirement. In our experiments, we vary this parameter within a certain range and show its influence.

The procedure of SMR is listed in Algorithm 1.

Algorithm 1 SMR

Input: Data matrix $\mathbf{X_i}$ for $i=1,2,\cdot,l$, labels $\{\mathbf{y}_1,\mathbf{y}_2,\cdots,\mathbf{y}_l\}$, balance parameter $\alpha>0$, rank parameter $k\in\{1,2,\cdots\}$, sparse constraint parameter $p\in(0,1]$.
Output: Selected feature index $\{r_1,r_2,\cdots,r_s\}$.
1: Initialize $\mathbf{V}=[\mathbf{I}_{k\times k},\mathbf{0}_{k\times(n-k)}]^T$ and $\mathbf{D}^{(v)}=\mathbf{I}$;
Repeat
2: Update $\hat{\mathbf{u}}^{(r)}$ and b_r using (23);
3: Update $\mathbf{D}^{(u)}$ using (29);
4: Update $\hat{\mathbf{v}}^{(r)}$ and b_r using (35);
5: Update $\mathbf{D}^{(v)}$ using (13);
Until converge
Feature Selection
6: Compute $\mathbf{P}$ using (25);
7: Compute the scores for all features using the 2-norm of the rows of $\mathbf{P}$;
8: Sort these scores and select the largest s values. The corresponding indexes form the selected feature index set $\{r_1,r_2,\cdots,r_s\}$. The corresponding elements of the input matrices are the selected features.

IV. Performance Analysis

A. Convergence Analysis

In previous section, we have solved SMR in an alternative way. The following proposition guarantees that our solving strategy can decrease the objective function of SMR shown in (11) in each iteration.

Proposition 1: In each iteration, the objective function value of SMR in (11) is non-increasing by employing the optimization procedure in Algorithm 1.

There are three key points for its proof. 1) Solving the approximated problem in (14) (or (28)) by the iterative strategy listed in Algorithm 1, we can decrease the objective function of SMR in (11). 2) Using the strategies in Algorithm 1, we can also see that the objective function in (14) is non-increasing, since in each iteration, we have reformulated it as a regularized least squares problem and the optimal solution can be derived in a closed form. 3) The reformulated problem in (27) has the same objective function values as that of the problem shown in (11). Due to the space limitation, we omit the details.

Besides, we would also like to analyze the computational complexity of SMR. In each iteration, we only need to solve two linear regression problems taking mk and nk dimensional data points as the inputs. A plenty of researches have been dedicated to solving the linear regression problem and there are many fast solving algorithms. For example, in [36] and [37], an effective method, named as LSQR, has been proposed for solving linear regression problem. Assume the linear regression model is $\mathbf{Ax}=\mathbf{b}$ with $\mathbf{A}\in\mathbb{R}^{m\times n}$, the computational complexity of LSQR is $O(t(2mn+3m+5n))$, where t is the number of iterations. As mentioned in [37], LSQR converges very fast. Besides, as in the related literatures in solving the $\ell_{2,p}$-norm regularization problem [30], the iteration often has fast convergence speed. For example, in the following experiments, SMR converges within 30 iterations. Comparing with traditional vector-based methods which take an mn-dimensional data as the input, our proposed SMR scales well in practice.

B. Parameter Determination

There are three parameters in SMR, i.e., α, k, and p. p can be determined as in previous work [26]. α is the parameter which balances the effects of matrix regression and sparse constraints for feature selection. k is the number of regression vectors. The following proposition reveals the essence of k.

Proposition 2: Assume $\{\mathbf{u}_1,\mathbf{u}_2,\cdots,\mathbf{u}_k\}$ are any k vectors of dimensionality m, $\{\mathbf{v}_1,\mathbf{v}_2,\cdots,\mathbf{v}_k\}$ are k vectors of dimensionality n. If $k=\min(m,n)$, then the dimensionality of space spanned by Vec$(\sum_i^k\mathbf{u}_i\mathbf{v}_i^T)$ is mn.

The proof of this result mainly includes two aspects. On the one hand, for any mn-dimensional vector $\mathbf{z}$, we can find k couples of vectors, such that $\mathbf{z} = \text{Vec}(\sum_i^k \mathbf{u}_i \mathbf{v}_i^T)$. On the other hand, for any two groups of vectors, denoted as $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\}$ and $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k\}$, $\text{Vec}(\sum_i^k \mathbf{u}_i \mathbf{v}_i^T) \in \mathbb{R}^{mn}$. We omit the proof due to the limitation of space.

As seen from this proposition, k is a parameter to determine the complexity of the model in essence. The larger k is, the more complicated this model is. The extreme cases are $k = 1$ and $k = \min(m, n)$. When $k = 1$, it is the Generalized Bilinear Regression (GBR) model. When $k = \min(m, n)$, SMR tends to be the traditional vector based regression model. In other words, the first objective of SMR can be regarded as the trade-off between these two models.

Different parameters play different roles in the model and it is difficult to determine the optimal parameters simultaneously by the same way. We determine them in a heuristic way by two dimensional grid search. Concretely, we vary both of them in the predefined ranges and select the parameters corresponding to the best performance. In the next section, we will give some experimental results to show their influence.

C. Computational Complexity

As seen from the procedure of SMR in Algorithm 1, the most computational steps of SMR are updating $\hat{\mathbf{u}}^{(r)}$, b_r using (23), and updating $\hat{\mathbf{v}}^{(r)}$, b_r using (35). In (23), it is the solution to regularized least squares problem in (22) and the computational complexity is $O(m^2k^2)$. Correspondingly, the computational complexity in (35) is $O(n^2k^2)$. Assume that the total number of iterations is T. The computational complexity of SMR is $O(\max\{m^2, n^2\} \times k^2 \times c \times T)$. Traditional vector based regression model takes $m \times n$ vector as the input and its computational complexity is $O(m^2n^2 \times c)$. Commonly, T is less than 30 and k is far less than $\min\{m, n\}$. Thus, the computational complexity of SMR is far less than traditional regression model.

V. Experiments

In this section, we compare SMR with other popular vector based feature selection approaches on several public data sets. We also provide several results for convergence behaviour, parameter determination and computational cost.

A. Data Description and Evaluation Metric

In our experiments, four public data sets are employed to show the performance of different feature selection methods. They are three face image data sets, including ORL, PIE[2] and Umist,[3] one handwritten digit data, i.e., USPS.[4] The statistics of the data sets are summarized in Table I.

To test the quality of selected features, we employ two different kinds of evaluation metrics, i.e., accuracy-the classification accuracy achieved by classifier using the selected features; redundancy rate (RED)-the redundancy rate contained in the selected features. Intuitively, an ideal feature selection approach should select features with high classification accuracy and few redundancy. The redundancy is a popular evaluation metric for feature selection. It measures the quality of selected features directly, without employing the following tasks. Assume $\mathcal{F}$ is the set of selected features and $\mathbf{X}_\mathcal{F}$ is the data represented by the features in $\mathcal{F}$. The following metric measures the redundancy rate of $\mathcal{F}$ [38]:

$$\text{RED}(\mathcal{F}) = \frac{1}{|\mathcal{F}|\,(|\mathcal{F}| - 1)} \sum_{\mathbf{s}_i, \mathbf{s}_j \in \mathcal{F}, i > j} \text{corr}_{i,j}, \qquad (36)$$

where $|\mathcal{F}|$ is the cardinality of $\mathcal{F}$ and $\text{corr}_{i,j}$ is the Pearson correlation coefficient between two features $\mathbf{s}_i$ and $\mathbf{s}_j$, computed by using the data points in $\mathbf{X}_\mathcal{F}$. This measurement assesses the averaged correlation among all feature pairs, and a large value indicates that many selected features are correlated and thus high redundancy is expected to exist in $\mathcal{F}$.

To compute the classification accuracy, we use the 1-Nearest Neighbourhood classifier (NN for simplicity) [3] to perform classification on the data with selected features. We randomly select a fixed number of examples from each category as training data and the rest are assigned as testing data. We perform feature selection on training data.

In the following, we compare SMR with six popular feature selection approaches. 1) Information Gain (IG) [27], which measures the number of bits of information obtained for prediction of a class by knowing the presence or absence of a feature. 2) ReliefF [29], which evaluates features based on how well the feature differentiates between neighboring instances from different classes versus from the same class. 3) Fisher Score (FisherScor) [3], which uses discriminative methods and generative statistical models to determine the most relevant features for classification. 4) Robust Feature Selection (RFS) [30], which selects features with robust regression manner. Concretely, the authors replace the ℓ_2-norm in traditional lease squares regression by the $\ell_{2,1}$-norm to enhance robustness of

TABLE I
Data Set Description

Data	# of points	# of features	# of classes	Type
ORL	400	32×32	40	Image, Face
PIE	11554	32×32	68	Image, Face
Umist	575	28×23	20	Image, Face
USPS	2007	16×16	10	Image, Digit

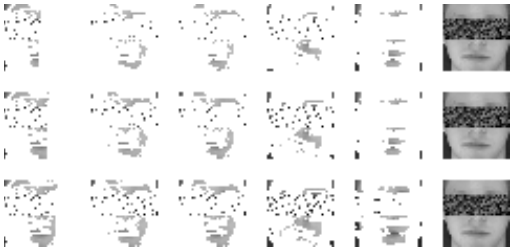


Fig. 3. Feature selection results on the occluded ORL image. From top to bottom, the number of selected features are 100, 150 and 200. From left to right, the methods are IG, ReliefF, FisherScor, RFS and SMR. The rightmost column contains the original image.

[2] http://vasc.ri.cmu.edu/idb/html/face/index.html
[3] http://researchweb.iiit.ac.in/~chetan/face-retrieval-php/
[4] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

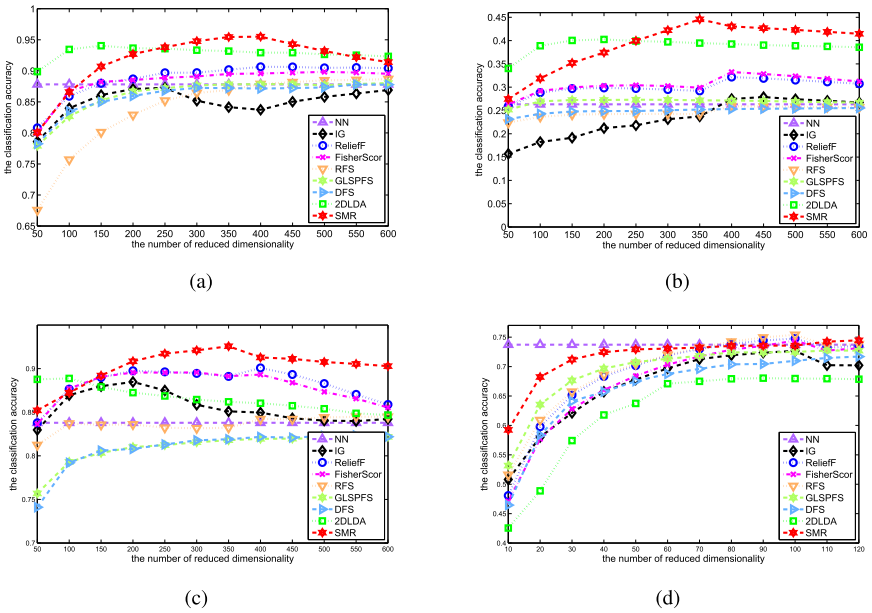


Fig. 4. Classification results of different methods on four different data sets with different numbers of selected features. (a) ORL. (b) PIE. (c) Umist. (d) USPS.

regression. 5) Global and Local Structure Preservation framework for Feature Selection (GLSPFS) [13], which integrates both global pairwise sample similarity and local geometric data structure to conduct feature selection. 6) Discriminative Feature Selection (DFS) [26], which combines the popular transformation-based dimensionality reduction method LDA and sparsity regularization for feature selection. Besides, we also compare our method with 2DLDA, which is the most popular supervised two dimensional feature extraction approach. Moreover, the results of NN are also reported as the baseline.

In the implementation of SMR, we set $p = 1$ when comparing it with other algorithms since the efficiency of the $\ell_{2,1}$-norm in feature selection has been demonstrated in many studies [5], [30]. Previous results show that the selection of p does not take a great influence on the sparse constraint, i.e., $\ell_{2,p}$, which is designed for feature selection [5]. For simplicity, the parameters, α and k, are determined by two dimensional grid search in a heuristic way. Concretely, the regularization parameters α are tuned from $\{0.001, 0.01, 0.1, \cdots, 1000\}$ and the k parameter is tuned from $\{2, 4, \cdots, 10\}$. The parameters in other algorithms are also tuned by grid search.

B. Toy Example

To show that SMR could use location information in feature selection, we add noise to the images from the first two classes of ORL data. The typical images are shown in Fig. 1. The eyes from each face are occluded by noise sampled from the uniform distribution in the range [0,0.6]. We compare SMR with other four vector based feature selection approaches and show the results in Fig. 3. From top to bottom, the number of selected features are 100, 150 and 200. From left to right, the methods are IG, RelieF, FisherScor, RFS, and SMR. The features, which are not selected, are shown in white. The rightmost column contains the original image. The parameter settings are the same as follows.

As seen from the results in Fig. 3, it is clear that SMR could use the block wise structure information by treating element in each row (or column) as a part. This type of occlusion can be detected and rarely selected. Other methods, which treat all the pixels equally, select the pixels from the noisy occlusion, since they only choose the features with more discriminative power and omit the location information. With the increase of selected feature number, there are more and more useless features (features from the occlusion), which are selected by other methods. Certainly, this is just a toy example. It is not guaranteed that SMR is totally resilient to this type of noise. For example, we cannot guarantee that SMR can tackle this type of noise at any level.

C. Comparison Between SMR and Other Algorithms

In this section, we provide some numerical results on four data sets using the above mentioned two metrics.

We report two groups of experiments. In the first group, we compare the classification accuracy of different methods. Since different data sets have different scales, we randomly select 5, 6, 4 and 4 data points per class from ORL, PIE, Umist and USPS data sets as training samples and the remaining data composes the test examples. For each data set, the total number of features is listed in Table I. We set the number of selected features between 50 and 600 with an interval 50 for all data sets except USPS since its data scale is small. Correspondingly, we set the reduced dimensionality of 2DLDA with the same numbers. Since the aim of feature selection is to find compact representation, we have constrained the number of selected features within a small range and not reported the results with all features. Each feature selection algorithm is first performed on training data to determine the selected features. Then, we train a classifier on training data with only

TABLE II
FEATURE REDUNDANCIES (RED) ON THE PIE AND USPS IRIS DATA SETS

dataset	s	IG	ReliefF	FisherScor	RFS	GLSPFS	DFS	SMR
PIE	50	0.6893±0.0001	0.4119±0.0441	0.4762±0.0651	0.4245±0.0235	0.4146±0.0435	0.4108±0.02738	**0.4079**±0.0213
	100	0.6912±0.0002	0.3972±0.0351	0.4766±0.0599	0.4159±0.0163	0.4126±0.0377	0.4141±0.0174	**0.4038**±0.0130
	150	0.6991±0.0001	0.3963±0.0345	0.4752±0.0533	0.4145±0.0110	0.4021±0.0317	0.4119±0.0158	**0.4009**±0.0134
	200	0.6963±0.0001	0.3995±0.0349	0.4721±0.0457	0.4115±0.0113	0.4026±0.0249	0.4069±0.0120	**0.3980**±0.0147
	250	0.6922±0.0002	0.4043±0.0355	0.4693±0.0400	0.4097±0.0097	0.4056±0.0249	0.4025±0.0086	**0.3939**±0.0127
	300	0.6650±0.0002	0.4087±0.0333	0.4674±0.0379	0.4093±0.0084	0.4061±0.0219	0.4042±0.0088	**0.3910**±0.0115
	350	0.6602±0.0001	0.4148±0.0316	0.4640±0.0355	0.4074±0.0077	0.4073±0.0202	0.4064±0.0079	**0.3904**±0.0103
dataset	s	IG	ReliefF	FisherScor	RFS	GLSPFS	DFS	SMR
USPS	10	0.4060±0.0962	0.2882±0.0695	0.4307±0.1101	0.4023±0.0741	0.2128±0.0493	0.2058±0.0312	**0.2010**±0.0216
	20	0.4059±0.0596	0.2586±0.0432	0.3794±0.0639	0.3685±0.0525	0.2130±0.0226	0.2083±0.0281	**0.2053**±0.0136
	30	0.3891±0.0546	0.2436±0.0352	0.3612±0.0620	0.3462±0.0438	0.2198±0.0188	0.2138±0.0170	**0.2098**±0.0083
	40	0.3599±0.0388	0.2379±0.0299	0.3399±0.0517	0.3305±0.0338	0.2188±0.0172	0.2190±0.0149	**0.2156**±0.0142
	50	0.3368±0.0355	0.2328±0.0255	0.3249±0.0431	0.3165±0.0286	0.2255±0.0172	0.2253±0.0131	**0.2208**±0.0139
	60	0.3153±0.0289	0.2296±0.0227	0.3114±0.0377	0.3043±0.0228	0.2275±0.0146	0.2221±0.0106	**0.2216**±0.0116
	70	0.2961±0.0302	0.2254±0.0191	0.3009±0.0334	0.2938±0.0195	0.2271±0.0137	0.2268±0.0111	**0.2242**±0.0105

the selected features. After that, we use the trained classifier to classify testing data with selected features. We repeat this procedure for 20 independent runs and the mean classification accuracies are the final results shown in Fig. 4.

In the second group, we also calculated the RED measure, defined in (36), between selected features. Since NN and 2DLDA are not feature selection methods, we omitted their results. The parameters are the same as that in the first group of experiments. With different training data, we have different rankings of all features. In each run, we select the most important s features to formulate the set $\mathcal{F}$ in (36) and compute the RED values. Here, $\text{corr}_{i,j}$ is computed using all data points. For each approach, the mean and standard derivation of 20 independent RED values are shown in Table II. Due to the limitation of space, we only report the results on two representative data sets. The smallest values, which indicate the best performances of feature selection, are boldfaced.

As seen from the results in Fig. 4, the classification accuracies of different feature selection approaches vary with the increase of the number of selected features. For data sets, such as PIE, USPS, with more selected features, all feature selection approaches seem to achieve higher classification accuracies. A similar tendency can also be found on another two data sets, with only IG's performance fluctuating. Nevertheless, its classification accuracy fluctuates within a certain range. This may be caused by the fact that we only select a small number of features. Besides, the above tendency does not mean that all methods perform better with more features. For example, the results of NN are not always the highest for all methods. Besides, with the increase of the number of selected features, the performance of all methods approach that of the baseline, with the NN classifier. It is consistent with intuition. When we selected all features, all the methods achieve the same results.

In terms of the classification accuracy, we have the following observations. 1) Comparing with other feature selection approaches, SMR outperforms all the other feature selection methods on all data sets in most of the time. For example, on the PIE data sets, compared to the best result of all the other methods, SMR gets about 7% improvements in average. On the USPS data, the average improvements are about 5%. In terms of the RED results, SMR also performs the best. The above mentioned improvements can also be seen from the results in Table II. On the PIE data, our method consistently performs better than all the other feature selection approaches. 2) Comparing with the baseline NN classifier, SMR outperforms NN in the first three data sets. On the USPS data set, the baseline with NN achieves the best results. It achieves higher accuracies than all the dimensionality reduction approaches, including all the feature selection approaches and feature extraction approaches, i.e., 2DLDA. The reason may be that the original representation of USPS data is compact enough and it is not necessary to reduce its dimensionality. 3) Comparing with 2DLDA, which is a representative supervised two dimensional feature extraction approach, SMR achieves comparable performance. Moreover, as show in Fig. 4, when the number of reduction dimensionality is large, SMR outperforms 2DLDA in all data sets. This may be caused by the reason that with a little more features, the original representation is good enough or even better than the translated formulation derived by feature extraction approach. 4) Comparing with the accuracies on other data sets, the classification accuracies of all methods on the PIE data set (Figure 4(b)) is quite low. It may be due to the fact that the number of categories of PIE is larger than other data sets. Besides, the number of testing examples is also much larger than other data sets.

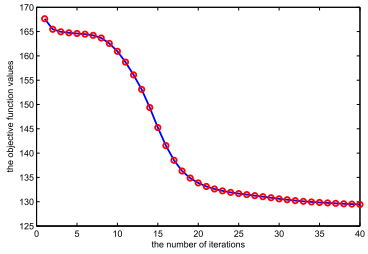
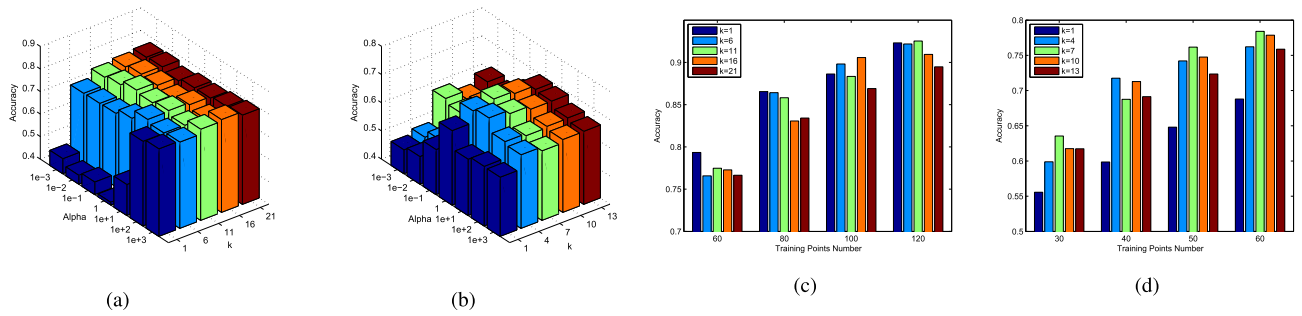


Fig. 5. Objective values of SMR with different numbers of iterations on the ORL data.



(a) (b) (c) (d)

Fig. 6. Classification accuracies of SMR with different parameters. (a) Classification accuracies on the Umist data with different α and k; (b) Classification accuracies on the USPS data with different α and k; (c) Classification accuracies on the Umist data with fixed α and different k; (d) Classification accuracies on the USPS data with fixed α and different k. We fixed α as the optimal parameter determined by grid search shown in (a) and (b).

TABLE III
COMPUTATIONAL TIME COMPARISON ON FOUR DATA SETS

dataset	IG	ReliefF	FisherScor	RFS	GLSPFS	DFS	2DLDA	SMR
ORL	0.47±0.07	1.76±0.12	1.85±0.05	1.21±0.02	0.95±0.18	45.93±0.43	0.15±0.13	40.36±0.47
PIE	0.70±0.08	4.71±0.17	2.40±0.26	3.42±0.31	4.17±0.28	38.14±1.10	98.37±1.61	68.44±2.65
Umist	0.26±0.06	0.43±0.03	0.63±0.06	0.24±0.01	0.22±0.05	12.36±0.52	0.16±0.03	7.50±0.29
USPS	0.11±0.03	0.11±0.04	0.17±0.08	0.05±0.03	0.04±0.01	1.17±0.07	0.41±0.08	1.12±0.10

D. Convergence and Parameters Determination

To show the convergence behaviour, we report the objective function values on the ORL data set, since the algorithm has similar convergence behaviour on other data sets. The convergence curves are displayed in Fig. 5. As seen from Fig. 5, the objective function values are non-increasing during the iterations and it converges to a fixed value.

As for parameter determination, the k parameter plays an important role in balancing the capacity of learning and generalization. The parameter α balances the effectiveness of regression and feature selection. In our paper, we use the grid search to determine them heuristically. Concretely, we randomly choose some training points as validation samples, vary the parameters within predefined sets and find the parameter combination that can achieve the highest classification accuracy on validation points. To show their influence, we present the classification accuracies of SMR with different combinations of k and α on the Umist and USPS data sets for illustration. The results are shown in Fig. 6(a) and (b). Besides, to show the effectiveness of multiple regression vectors, we also want to show the unique influence of k. A group of results with fixed α and various k is presented. With different numbers of training points, we vary k with a large range(k should be within the range $[1, \min\{m, n\}]$) and show the results in Fig. 6(c) and (d).

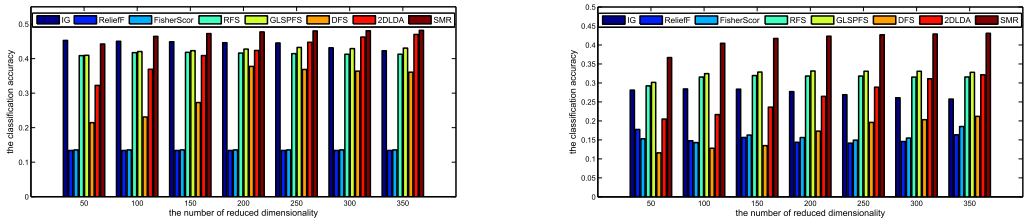
As seen from Fig. 6(a) and (b), with different combinations of k and α, the performance of SMR varies. In our experiments, since the parameters k and α change within large ranges, the performances of SMR also changes drastically. It demonstrates that the two parameters are vital to dominate the performance of SMR. Besides, for higher values of α, it seems that the chosen value of k it is not important. It may be caused by the fact that, with large α, we will emphasize on the regularization term in (10). The optimal solution to (10) is all zeros and the influence of k will be weak. It is still a challenging problem to find suitable parameter setting for specific data. Thus, in real applications, we run the algorithm in different parameter settings and choose the best one. Besides, as shown in Fig. 6(c) and (d), with the increase of k, the fitting error decreases. However, the classification accuracy does not always increase consistently. This is due to the fact that k is a parameter to balance the influences of training error and capacity of generalization. They also validate the effectiveness in using multiple regression vectors.

E. Computational Time Comparison

In this section, we will compare the computational time of different methods. We conduct experiments on above mentioned data sets. For illustration, we compare SMR with IG, ReliefF, FisherScor, RFS, GLSPFS and DFS. Since NN is evaluation method and all the other approaches employ them for classification, we have not compared with them. For justice, these methods are all implemented in their original formulation, without using other accelerating strategies. Similar to the setting in Fig. 4, we fixed number of training points and randomly select training points for 50 runs. With a naive MATLAB implementation, the calculations are made on a 2.7-GHz Windows machine. The CPU time of different methods are listed in Table III

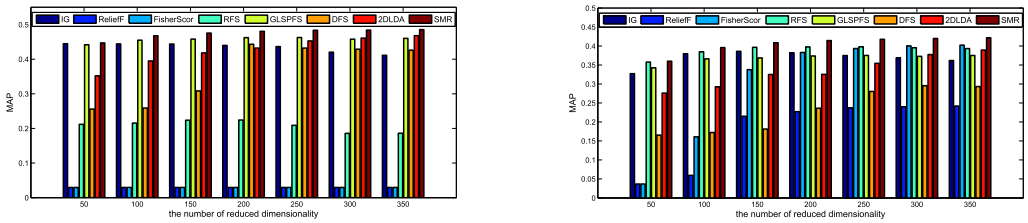
There are also some observations from these results. 1) Among different methods on different data sets, IG consumes the least time. Since SMR needs to made iterations, it costs more time. Nevertheless, the final output of SMR is the ranking of features. The little variance of objective function values, caused by the litter variance of optimal variables, takes small influence on the ranking of features. In most cases, the final feature ranking will not change after a few iterations. Thus, to save time, we can make a quick stopping criterion by measuring the difference of two adjacent ranking.

Fig. 7. Representative examples from OT and LSP data sets. The first five categories contain examples from OT and the last three columns are additional examples of LSP.



(a) (b)

Fig. 8. Classification accuracies of different methods on two scene classification tasks with different numbers of selected features. (a) OT. (b) LSP.



(a) (b)

Fig. 9. The MAP results of different methods on two scene classification tasks with different numbers of selected features. (a) OT. (b) LSP.

2) The computational time of different methods is dominated by different factors. For example, the dimensionality is the key factor in dominating the computational time of DFS and our method, whereas the number of points takes great influence on the computational efficiency of 2DLDA.

VI. Applications to Scene Classification

As a major open challenge in computer vision, scene classification (e.g., indoor, mountain, etc.) is an important task in visual understanding. It tries to represent the highest level concept (scene class) that an image depicts. Scene classification has received a lot of attention since the birth of this subject in computer vision [1]. We will test our feature selection approach in this challenging application scenario.

The common procedures of our experiments are listed as follows. 1) As in traditional scene classification approaches [1], [39], [40] we use the well-known feature descriptor to characterize each image. We do not employ our method on the raw features of original images since the images in this task are complicated and the raw features are not good enough to describe them. Previous works have also shown that taking the raw features as the input will degrade the performances [1], [41]. In our approach, we first use the four-level pyramid model to segment original images into 1, 4, 16, and 64 patches, respectively. In each patch, we use the 128-dimensional SIFT features, which are extracted from the particular interest points in the images. They are local and invariant to image scale and rotation, and robust to changes in illumination, noise, and minor changes in viewpoint. Consequently, the description of each image is a $128 \times 85(1 + 4 + 16 + 64)$ matrix. This matrix is then employed as the input of our algorithm. 2) We use different feature selection models to select the most important features. 3) After feature selection, all the images are represented by low dimensional vectors. As in previous experiments, we randomly separate the data into training and testing. The NN classifier is employed for scene classification.

Two commonly used data sets for scene classification, i.e., OT [39] and LSP [40], are employed for evaluation. OT contains 2,688 images from eight categories: 360 coasts, 328 forest, 374 mountain, 410 open country, 260 highway,

308 inside of cities, 356 tall buildings, and 292 streets. The average resolution of each image is 250×250 pixels. LSP includes 15 categories and is only available in gray scale. It consists of the 2,688 images (eight categories) of the OT data set plus: 241 suburb residence, 174 bedroom, 151 kitchen, 289 living room, 216 office, 315 store, and 311 industrial. The average resolution of each image is approximately 250×300 pixels. To show them intuitively, we have selected several images and presented them in Fig. 7. The first five categories are original images from OT data sets and the last three columns contain images from LSP solely. As seen from the example images, the scene classification task on LSP is more difficult since the backgrounds are more complicated.

Besides classification accuracy, as in the literature, we also employ another popular metric, i.e., Mean Average Precision (MAP), for evaluation [41]. MAP is commonly used for evaluating ranked lists in Information Retrieval. Suppose we are building a system to classify images of scene ¡°auditorium¡±. Let the system be evaluated on a test set. A good classifier is good at ranking actual ¡°auditorium¡± images near the top of the list. The performance of the classifier is measured by a single number called as Average Precision (AP). For the multi-class classification problem, the performance can be measured by the mean of AP (MAP) values of the individual classifiers.

In our experiments, the other settings are the same as in previous experiments. The classification accuracy and MAP comparison results are shown in Fig. 8 and Fig. 9, respectively. As we can see from the results in Fig. 8 and Fig. 9, it is obvious that SMR outperforms other feature selection approaches in most cases on the task of scene classification, no matter which kind of data sets and metrics we have employed. It indicates that our method can select the most discriminative features from different levels of the pyramid model and different directions of SIFT features. The reason may be the same as in previous experiments, i.e., considering spatial correlations and leveraging learning mechanism.

Interestingly, when the number of selected features is increasing within our predefined range, the improvement is more significant. This may be caused by the reason that a small number of features are not enough for distinguishing images from different categories. The performances of all methods degrade when the number of selected features is small. Moreover, due to the differences in data characteristic, the performances of different methods vary on different data sets. For example, RFS performs well on LSP data, while its performance is not so good on OT data. Besides, since LSP is the extension of OT and the task on LSP is more complicated than that on OT, all methods seem to achieve better results on the OT data set.

VII. Conclusion

In this paper, we aim to provide insights into the feature selection for matrix, or tensor data, directly, as well as to facilitate the design of new tensor algorithms. A novel algorithm named as SMR has been proposed. Sparse constrains are designed for feature selection. As illustrated in this paper, SMR has been shown to be more effective in selecting features for matrix data. Moreover, this algorithm can be extended to high order tensors directly. A byproduct of this paper is a series of theoretical analysis and some interesting optimization strategies. One of our future works is to systematically compare all possible extensions of the algorithms developed by different configurations of r and p in sparse regularization term, including its theoretical analysis and solving strategies. Another open problem is the selection of parameter k and α, which is an unsolved problem in many learning algorithms. In this paper, they are empirically determined. Additional analysis with different classifiers is also needed for this topic.

References

[1] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification using a hybrid generative/discriminative approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 712–727, Apr. 2008.

[2] D. Tao, X. Li, X. Wu, W. Hu, and S. J. Maybank, "Supervised tensor learning," *Knowl. Inf. Syst.*, vol. 13, no. 1, pp. 1–42, 2007.

[3] D. G. Stork, P. E. Hart, and R. O. Duda, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2000.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag, 2006.

[5] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 793–804, Jun. 2014.

[6] L. van der Maaten, E. Postma, and H. van den Herik, "Dimensionality reduction: A comparative review," Tilburg Centre Creative Comput., Tilburg Univ., Tilburg, The Netherlands, Tech. Rep. TiCC-TR 2009-005, 2009.

[7] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[8] J. Ye and Q. Li, "LDA/QR: An efficient and effective dimension reduction algorithm and its theoretical foundation," *Pattern Recognit.*, vol. 37, no. 4, pp. 851–854, 2004.

[9] D. Tao, X. Li, S. J. Maybank, and X. Wu, "Human carrying status in visual surveillance," in *Proc. CVPR*, 2006, pp. 1670–1677.

[10] Z. Lai, W. K. Wong, Y. Xu, C. Zhao, and M. Sun, "Sparse alignment for robust tensor learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1779–1792, Oct. 2014.

[11] J. Ye, R. Janardan, and Q. Li, "Two-dimensional linear discriminant analysis," in *Proc. NIPS*, 2004, pp. 1569–1576.

[12] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 212–220, Jan. 2007.

[13] X. Liu, L. Wang, J. Zhang, J. Yin, and H. Liu, "Global and local structure preservation for feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1083–1095, Jun. 2013.

[14] L. Wang, N. Zhou, and F. Chu, "A general wrapper approach to selection of class-dependent features," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1267–1278, Jul. 2008.

[15] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, Jan. 2004.

[16] J. Ye, "Generalized low rank approximations of matrices," in *Proc. ICML*, 2004, p. 112.

[17] D. Zhang and Z.-H. Zhou, "2D PCA: Two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, nos. 1–3, pp. 224–231, 2005.

[18] S. Noushath, G. H. Kumar, and P. Shivakumara, "2D LDA: An efficient approach for face recognition," *Pattern Recognit.*, vol. 39, no. 7, pp. 1396–1400, 2006.

[19] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in *Proc. NIPS*, 2006, pp. 499–506.

[20] J. Yang, D. Zhang, Y. Xu, and J. Yang, "Two-dimensional discriminant transform for face recognition," *Pattern Recognit.*, vol. 38, no. 7, pp. 1125–1129, 2005.

[21] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang, "Discriminant analysis with tensor representation," in *Proc. CVPR*, 2005, pp. 526–532.

[22] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.
[23] Y. Gao, X. Wang, Y. Cheng, and Z. Wang, "Dimensionality reduction for hyperspectral data based on class-aware tensor neighborhood graph and patch alignment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1582–1593, Aug. 2015.
[24] J. Sun, D. Tao, and C. Faloutsos, "Beyond streams and graphs: Dynamic tensor analysis," in *Proc. SIGKDD*, 2006, pp. 374–383.
[25] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recognit.*, vol. 44, no. 7, pp. 1540–1551, 2011.
[26] H. Tao, C. Hou, F. Nie, Y. Jiao, and D. Yi, "Effective discriminative feature selection with nontrivial solution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 796–808, Apr. 2016.
[27] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Hoboken, NJ, USA: Wiley, 2006.
[28] 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, no. 8, pp. 1226–1238, Aug. 2005.
[29] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Proc. 10th Nat. Conf. Artif. Intell.*, 1992, pp. 129–134.
[30] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint ℓ2,1-norms minimization," in *Proc. NIPS*, 2010, pp. 1813–1821.
[31] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. San Mateo, CA, USA: Morgan Kaufmann, 2011.
[32] J. Li *et al.*, (2016). "Feature selection: A data perspective." [Online]. Available: https://arxiv.org/abs/1601.07996
[33] K. R. Gabriel, "Generalised bilinear regression," *Biometrika*, vol. 85, pp. 689–700, Sep. 1998.
[34] C. Hou, F. Nie, D. Yi, and Y. Wu, "Efficient image classification via multiple rank regression," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 340–352, Jan. 2013.
[35] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.
[36] C. C. Paige and M. A. Saunders, "Algorithm 583: LSQR: Sparse linear equations and least squares problems," *ACM Trans. Math. Softw.*, vol. 8, no. 2, pp. 195–209, Jun. 1982.
[37] C. C. Paige and M. A. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Trans. Math. Softw.*, vol. 8, no. 1, pp. 43–71, Mar. 1982.
[38] Z. Zhao, F. Morstatter, S. Sharma, S. Alelyani, A. Anand, and H. Liu, "Advancing feature selection research-asu feature selection repository," School Comput. Informat., Decision Syst. Eng., Arizona State Univ., Tempe, AZ, USA, Tech. Rep. TR-10-007, 2010.
[39] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.
[40] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. CVPR*, 2006, pp. 2169–2178.
[41] Y. Huang, Z. Wu, L. Wang, and T. Tan, "Feature coding in image classification: A comprehensive study," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 493–506, Mar. 2014.

Chenping Hou (M'12) received the B.S. and Ph.D. degrees in applied mathematics from the National University of Defense Technology, Changsha, China, in 2004 and 2009, respectively. He is currently an Associate Professor with the College of Science, National University of Defense Technology. He has authored several papers in journals and conferences, such as the IEEE TNNSL, IEEE TCYB, IEEE TIP, Pattern Recognition, IJCAI, and AAAI. He has served a PC Members for NIPS, IJCAI, AAAI, and ICASP. His current research interests include pattern recognition, machine learning, data mining, and computer vision.

Yuanyuan Jiao received the Ph.D degree in applied mathematics from the National University of Defense Technology, Changsha, China, in 2012. Her research interests include data mining and its applications.

Feiping Nie received the Ph.D. degree in computer science from Tsinghua University, China, in 2009. His research interests are machine learning and its application fields, such as pattern recognition, data mining, computer vision, image processing, and information retrieval. He has authored over 100 papers in the following top journals and conferences. According to the Google scholar, his papers have been cited over 2000 times. He is now serving as an Associate Editor or a PC member for several prestigious journals and conferences.

Tingjin Luo received the B.S. and master's degrees with the College of Information System and Management, National University of Defense Technology, Changsha, China, in 2013 and 2011, respectively, where he is currently pursuing the Ph.D. degree with the College of Science. His research interests include machine learning, multimedia analysis, and computer vision.

Zhi-Hua Zhou (S'00–M'01–SM'06–F'13) received the B.Sc., M.Sc., and Ph.D. degrees (Hons.) in computer science from Nanjing University, China, in 1996, 1998, and 2000, respectively. He joined the Department of Computer Science and Technology, Nanjing University, as an Assistant Professor in 2001, and is currently a Chair Professor and Standing Deputy Director of the National Key Laboratory for Novel Software Technology. He is also the Founding Director of the LAMDA Group. His research interests are mainly in artificial intelligence, machine learning, and data mining. He has authored the books Machine Learning (in Chinese) and Ensemble Methods: Foundations and Algorithms, and published over 100 papers in top-tier international journals or conference proceedings. He is a fellow of the ACM, AAAI, AAAS, IAPR, IET/IEE, and CCF. He has received various awards/honors including the National Natural Science Award of China, the PAKDD Distinguished Contribution Award, the IEEE ICDM Outstanding Service Award, the IEEE CIS Outstanding Early Career Award, and the Microsoft Professorship Award. He also holds 16 patents. He is an Executive Editor-in-Chief of Frontiers of Computer Science, Associate Editor-in-Chief of Science China: Information Sciences, Action Editor of Machine Learning, and Associate Editor of ACM Transactions on Intelligent Systems and Technology and the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He served as an Associate Editor-in-Chief for Chinese Science Bulletin, Associate Editor or Editorial Board Member for the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, and Artificial Intelligence in Medicine, Knowledge and Information Systems, Neural Networks. He was the Founder of the Asian Conference on Machine Learning and serves for the Steering Committee of the ACML, PAKDD, PRICAI, and IEEE ICDM. He served as the Advisory Committee Member of IJCAI 2015–2016, the General Chair/Co-Chair of ACML 2012, PAKDD 2014, and ICDM 2016, and the Program Committee Chair/Co-Chair of SDM 2013, ICDM 2015, IJCAI 2015, and Machine learning Track. He frequently served as an Area Chair of the AAAI, IJCAI, ICML, NIPS, KDD, and ICDM. He is the Chair of the CCF-AI and the Vice-Chair of the IEEE Nanjing Section. He served as the Chair of the IEEE CIS Data Mining Technical Committee, the IEEE Computer Society Nanjing Chapter, and the CAAI Machine Learning Technical Committee.