Marginal Representation Learning with Graph Structure Self-Adaptation

Zheng Zhang, Ling Shao, *Senior Member, IEEE*, Yong Xu, *Senior Member, IEEE*, Li Liu, Jian Yang

***Abstract*—Learning discriminative feature representations has shown remarkable importance due to its promising performance for machine learning problems. This paper presents a discriminative data representation learning framework by employing a simple yet powerful marginal regression function with probabilistic graphical structure adaptation. A marginally structured representation learning (MSRL) method is proposed by seamlessly incorporating distinguishable regression targets analysis, graph structure adaptation and robust linear structural learning into a joint framework. Specifically, MSRL learns marginal regression targets from data rather than exploiting the conventional zero-one matrix that greatly hinders the freedom of regression fitness and degrades the performance of regression results. Meanwhile, an optimized graph regularization term with self-improving adaptation is constructed based on probabilistic connection knowledge to improve the compactness of the learned representation. Additionally, the regression targets are further predicted by utilizing the explanatory factors from the latent subspace of data, which can uncover the underlying feature correlations to enhance the reliability. The resulting optimization problem can be elegantly solved by an efficient iterative algorithm. Finally, the proposed method is evaluated by eight diverse but related tasks, including object, face, texture, and scene categorization datasets. The encouraging experimental results and the explicit theoretical analysis demonstrate the efficacy of the proposed representation learning method in comparison with state-of-the-art algorithms.**

***Index Terms*—Discriminative representation, low-rank representation, sparse representation, block-diagonal structure, image recognition.**

I. Introduction

Learning discriminative and effective visual representations of data makes extracting informative features easier when constructing classifiers or other predictors. It is worth noting that learning good data representations is beneficial to the machine learning community, and an effective data representation can disentangle the subtle but important underlying information hidden in the observed data [1]. In addition to deep representations [1], there are extensive data representation learning methods, such as manifold-inspired representations [2]–[4], sparse and low-rank representations [5]–[10], and dictionary representations [11]–[13].

It is known that high-dimensional data usually contain certain redundant, irrelevant and noisy information, which greatly hamper the use of a designed system. Manifold-inspired representation learning mainly considers to determine a lower-dimensional but the most expressive subspace. It identifies a subset of significant features ($i.e.$ feature selection) or finds a projection or a transformation to approximate all features with a certain criterion ($i.e.$ feature extraction) [2]–[4], [14]. Moreover, some linearized versions of subspace learning algorithms can overcome the out-of-sample problem [15]. However, computational infeasibility becomes one of the main obstacles when applying them to a large number of local features [16]. Additionally, learning desirable graph topology of data plays a critical role in the success of graph-based representations [17].

The research on sparse and low-rank representations [5], [7], [8] has gained considerable attention due to their promising performance when solving different computer vision problems. For example, sparse representation classifier (SRC) [5], [6] has been successfully applied to robust face recognition. The nature of sparse representation learning is to select the most discriminative representations from the observed data and shrink the others. Some efficient representation learning methods were proposed, such as collaborative representation based classification (CRC) [18], locality-constrained linear coding algorithm (LLC) [19], and linear repression classifier (LRC) [20]. Instead of using the l_1-norm minimization, they employ the l_2-norm regularization to achieve the compromised goal that is computationally efficient without compressing the performance. However, sparsity-inferred methods may be incapable of capturing the global structure of data because these algorithms are dedicated to searching the sparsest representation of each sample individually. Such limitation has led to the emergence of the low-rank representation [8]–[10]. The low-rank based methods study the representation that jointly uncovers the underlying correlations between samples and globally preserves the membership of data. Robust principal component analysis (RPCA) [21] is one of the most representative low-rank constrained methods. To generalize the low-rank property to handle data from multiple subspaces, a low-rank representation based method (LRR) [8] was introduced to make subspace segmentation. Due to its simplicity and effectiveness, a lot of LRR-based algorithms have been developed, such as latent LRR (LatLRR) [22], Laplacian regularized LRR [23] and low-rank ridge regression (LRRR) [24].

Manuscript received May 26, 2017; revised *** **, 2017; accepted Nov. 6, 2017. This work was partially supported by the National Natural Science Foundation of China under Grant 61233011, and Guangdong Province high-level personnel of special support program (No. 2016TX03X164). (*Corresponding author: Yong Xu.*)

Z. Zhang and Y. Xu are with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, P. R. China (e-mail:darrenzz219@gmail.com; yongxu@ymail.com).

L. Shao and L. Liu are with the School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, U.K. (e-mail: ling.shao@ieee.org, liuli1213@gmail.com).

Jian Yang is with the College of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, P. R. China (e-mail: csjyang@njust.edu.cn).

The main objective of dictionary learning (DL) is to learn more compact data representations adapted to various tasks from the original data under certain criteria. For example, K-SVD [11] is one of the most well-known DL algorithms for signal restoration and denoising. However, K-SVD may be ineffective in classification or regression, as it mainly concentrates on reformulating an overcomplete dictionary that best reconstructs the input data, ignoring any discriminating metric such as the subspace and label information. To accommodate machine learning tasks, a series of DL methods have been developed to improve the discriminative ability of the learned dictionary to deal with labeled data, such as discriminative K-SVD (D-KSVD) [12], label consistent D-KSVD (LC-KSVD) [13] and locality constrained and label embedding dictionary learning (LCLE-DL) [25]. Moreover, DLSI [26] learns each sub-dictionary individually by imposing the structural incoherent information between classes, and CBDS [27] constructs a class-wise block-diagonal structure for discriminative dictionary learning. Linearized kernel DL [28] composed of kernel matrix approximation and virtual sample construction easily formulates some existing supervised and unsupervised DL algorithms to their kernel versions.

However, existing representation learning algorithms are not flexible and adaptive enough for machine learning tasks such as recognition or regression. The main limitations of these methods are fragility to the presence of outliers, computational infeasibility and weak discriminability. Specifically, manifold representations are learned by refining or projecting the original data to a new subspace, but overcoming high computation and finding the desirable graph structure become two challenging topics. It is known that the conventional sparse and low-rank representations cannot satisfy the demanding needs of the real-time applications due to heavy computational burden [5], [7], [8], [18], [21], [22]. Moreover, the learned data representations still lack distinguishable capabilities of capturing the potential explanatory factors for the observed input from different subjects. To remedy these deficiencies, this paper proposes a marginally structured representation learning (MSRL) method for efficient and effective visual representation learning. First, our MSRL algorithm is mainly based on a simple but effective marginal regression targets learning. Instead of utilizing the fixed zero-one matrix as regression targets, MSRL directly constructs self-tuning regression targets with a preferable near-optimal margin constraint. The regression results are more accurately measured. The probabilistic graphical structure adaptation is developed to capture underlying structures with *data connectivity*, which in turn guides the construction of marginal regression targets. In addition, the regression results are further predicted in the discriminative latent subspace of data, which can capture the underlying correlation patterns. The resulting formulation has the close-form solutions with respective to each subproblem, and can be elegantly solved by an efficient iterative algorithm. MSRL can also easily be extended to the semi-supervised version. Extensive experiments demonstrate the discrimination and effectiveness of the learned visual representations when solving different recognition tasks. In summary, the main contributions of the proposed MSRL framework are as follows:

(1) We propose to formulate a novel marginal visual representation learning framework based on joint flexible self-tuning marginal targets analysis, discriminative latent subspace construction and probabilistic graph structure adaptation. Therefore, the resulting data representations have obvious discriminative capabilities with the near-optimal margins, and our proposed methods achieve encouraging recognition results.

(2) The adaptive graph structure learning captures the probabilistic connectivity between each pair of samples in the regression task. The inherent structures of data are generally estimated by exploiting the shared information from data. Furthermore, the linear structural predictor learning employs the original and latent correlated information of data to make reliable predictions of regression task.

(3) Theoretical and experimental analyses of the convergence property for the optimization algorithm are explicitly presented, and the relationships between the proposed algorithms and several well-known algorithms are explicitly discussed. The extended semi-supervised version of MSRL is also introduced, and the efficiency of MSRL is further verified by the comparisons of the computational time.

The rest of this paper is organized as follows. We briefly introduce some related works in Section II. Then, the proposed MSRL method and theoretical analysis are described in Section III, and Section IV gives the optimization algorithm. Extensive experimental results are reported in Section VI, and the conclusion remarks are presented in Section VII.

II. Related Work

We first give some notations used in this paper. Matrices are denoted by bold uppercase letters, e.g. $\boldsymbol{X}$, and the i-th row and the j-th column element of matrix $\boldsymbol{X}$ is denoted as $\boldsymbol{X}_{ij}$. The bold lower letters indicate column vectors, $e.g.$ $\boldsymbol{x}$. The Frobenius norm of matrix $\boldsymbol{X}$ is defined as $\|\boldsymbol{X}\|_F^2 = tr(\boldsymbol{X}^T\boldsymbol{X}) = tr(\boldsymbol{X}\boldsymbol{X}^T)$, where $tr(\bullet)$ is the trace operator. $\|\boldsymbol{X}\|_*$ is the nuclear norm of matrix $\boldsymbol{X}$, $i.e.$ $\|\boldsymbol{X}\|_* = \sum_i |\sigma_i|$ where σ_i is the i-th singular value of matrix $\boldsymbol{X}$. The transposed matrix $\boldsymbol{X}$ is denoted as $\boldsymbol{X}^T$, and $\boldsymbol{I}$ denotes an identity matrix.

Due to its efficiency and effectiveness for data analysis, least squares regression (LSR) has been extensively used in many machine learning tasks. A series of improved algorithms of LSR have been proposed, such as discriminative LSR (DLSR) [29], discriminative elastic-net regularized LSR [30] and retargeted LSR [31]. Many popular models are also closely related to the original LSR model, such as sparse coding [5], [7], ridge regression [18], [20] and linear SVM [32], [33]. The conventional regularized LSR aims at learning a regression matrix $\boldsymbol{W} \in \Re^{d\times c}$, which projects the training samples $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n] \in \Re^{d\times n}$ to the label matrix $\boldsymbol{Y} \in \Re^{c\times n}$ by optimizing

$$\min_{\boldsymbol{W},\boldsymbol{b}} \|\boldsymbol{W}^T\boldsymbol{X} - \boldsymbol{b}\boldsymbol{e}_n^T - \boldsymbol{Y}\|_F^2 + \lambda\|\boldsymbol{W}\|_F^2, \tag{1}$$

where $\boldsymbol{b}$ is the regression error, $\boldsymbol{e}_n = [1, \cdots, 1]^T$ is a vector with all 1s, and λ is the regularization parameter. d, c, and n are respectively the dimension of sample, the number of classes and the number of samples. The j-th column of matrix

$\boldsymbol{Y}$, *i.e.* $\boldsymbol{y}_i = [0, \cdots, 0, 1, 0, \cdots, 0]^T \in \Re^c$, is the one-hot label vector of the i-th sample from the j-th class.

We can see that the regression targets $\boldsymbol{Y}$ of Eqn. (1) is a binary matrix. For the i-th row of $\boldsymbol{Y}$, *i.e.* $\boldsymbol{y}^i \in \{0, 1\}^n$, the elements corresponding to the data from the i-class are $1s$, otherwise $0s$. However, the zero-one target matrix is too rigid to make accurate regression. DLSR [29] utilizes the ε-dragging technique to force the binary outputs of different classes far away along opposite directions. The objective function of DLSR is

$$\begin{aligned} &\min_{\boldsymbol{W},\boldsymbol{b},\boldsymbol{M}} \|\boldsymbol{W}^T\boldsymbol{X} - \boldsymbol{b}\boldsymbol{e}_n^T - \boldsymbol{Y} - \boldsymbol{B} \odot \boldsymbol{M}\|_F^2 + \lambda\|\boldsymbol{W}\|_F^2, \\ &\quad s.t.\ \boldsymbol{M} \geq \boldsymbol{0}, \end{aligned} \tag{2}$$

where $\boldsymbol{B} \in \Re^{c\times n}$ is a constant matrix. If the j-th sample is from the i-th class, $\boldsymbol{B}_{ij} = +1$, otherwise $\boldsymbol{B}_{ij} = -1$. It is notable that DLSR [29] relaxes the binary matrix to a flexible matrix $\boldsymbol{R} = \boldsymbol{Y} + \boldsymbol{B} \odot \boldsymbol{M}$. Meanwhile, the margins between different classes are implicitly enlarged.

However, excessively pursuing the largest margins and greedily searching the best projection to fit the targets may lead to over-fitting with a high possibility. The graph-based learning technique [4], [14]–[16] provides a feasible approach to overcome the problem of over-fitting [34]. A significant advantage of graph structure learning is its capability to naturally capture diverse models of information or measurements, such as the similarity relationship of data [35]. The flexible graph-structure learning has been demonstrated its effectiveness in image clustering [35]. Nie et al. [36] proposed to integrate the stages of similarity matrix construction and unsupervised feature selection into a unified formulation. In many cases, it is beneficial to add a prominent graphical regularization term in the objective function of learning models. In this paper, a marginal visual representation learning method is proposed by constructing a simple but powerful discriminative regression model with the linear structural predictors learning and constructing favorable similarity relationships with optimized probabilistic graph structure adaption.

III. The Proposed MSRL framework

In this section, we present a novel marginal visual representation learning framework for image understanding, named marginally structural representation learning (MSRL), in which the discriminative high-level semantic information of data is extracted from low-level observed data.

The performance of machine learning methods is heavily dependent on data representations. The essential semantic information is always formulated by the combinations of useful representations, which are more effective and favorable than individual features. In light of this, the proposed marginal visual representation learning model can be stated in the following formula:

$$\min_{\boldsymbol{r}_i, f} \sum_{i=1}^{n} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{r}_i, f) + \beta\Psi(f) + \lambda\Phi(f), \tag{3}$$

where $\mathcal{L}(\cdot)$ is the discriminative loss function measuring the approximate regression error between the predefined marginalized targets and the prediction results, $\Psi(f)$ is the structural predictor learning to control the complexity of f, and $\Phi(f)$ is the adaptive graph learning to control the smoothness of function f. Additionally, λ and β are two hyper-parameters to balance the importance of the three terms.

A. The Loss Function

The choice of loss function $\mathcal{L}(\cdot)$ empirically depends on applications. There are three popular convex metric functions for recognition, the least squares loss, logistic loss and hinge loss. Among them, the simplest and most commonly used loss function is the least squares loss function, because it is both convex and smooth, and also can provide competitive performance to the hinge loss metric function in most cases. On the other hand, the logistic loss function is sensitive to outliers in comparison with the least squares loss metric. As a result, a tractable optimization problem based on the empirically squared loss function is formulated as

$$\mathcal{L}(\boldsymbol{x}_i, \boldsymbol{r}_i, f) = \sum_{i=1}^{n} \|f(\boldsymbol{x}_i) - \boldsymbol{r}_i\|_2^2, \tag{4}$$

where $\boldsymbol{r}_i \in \Re^c$ is the learned regression target.

The conventional LSR target vector is the label vector like $\boldsymbol{y}_i$ in Eqn. (1). However, taking the zero-one vector as the regression target is too tight to provide enough space to fit the regression problem. It is difficult to fully comply with a zero-one vector, which leads to high regression error [29]. To overcome this deficiency, we propose to directly learn the regression targets from data, and relax them to a flexible but marginal space in a supervised manner. Specifically, when optimizing the regression targets, we enforce a marginal constraint that the distance between the regression targets of the true and false classes should be larger than C, *i.e.*

$$\boldsymbol{r}_{il_i} - max_{j\neq l_i}\boldsymbol{r}_{ij} \geq C, \tag{5}$$

where C is a constant, and l_i indicates the position of the true class for the i-th sample. For example, if the i-th sample from the k-th class (*i.e.* $l_i = k$), the k-th element of the regression target $\boldsymbol{r}_i$, *i.e.* $\boldsymbol{r}_{ik}$, is bigger than the rest of elements by the margin of C. This simple learning trick can explicitly reflect the separability of each sample such that the margins of regression targets are enlarged and the inter-class separation is enhanced. Moreover, the linear transformation $\boldsymbol{W}$ is utilized as the mapping from the original data space $\Re^d$ to the marginal target space $\Re^c$:

$$\begin{aligned} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{r}_i, f) &= \sum_{i=1}^{n} \|\boldsymbol{W}^T\boldsymbol{x}_i - \boldsymbol{r}_i\|_2^2 \\ s.t.\ &\boldsymbol{r}_{il_i} - max_{j\neq l_i}\boldsymbol{r}_{ij} \geq C. \end{aligned} \tag{6}$$

B. The structural predictor learning

Reliable regularization is also an indispensable component for robust data representation as well as computational efficiency. To mine the underlying classes' correlation patterns, the low-rank regularization is imposed. More specifically, the

rank of $\boldsymbol{W}$ is explicitly determined by $s < min(c, n)$ and the following optimization problem is constructed:

$$\begin{aligned}&\min_{\boldsymbol{W},\boldsymbol{r}_i}\sum_{i=1}^{n}\|\boldsymbol{W}^T\boldsymbol{x}_i-\boldsymbol{r}_i\|_2^2\\&s.t.\ rank(\boldsymbol{W})\le s,\ \boldsymbol{r}_{il_i}-max_{j\neq l_i}\boldsymbol{r}_{ij}\ge C,\end{aligned}\tag{7}$$

which can be rewritten as

$$\begin{aligned}&\min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{r}_i}\sum_{i=1}^{n}\|(\boldsymbol{A}\boldsymbol{B})^T\boldsymbol{x}_i-\boldsymbol{r}_i\|_2^2\\&s.t.\ \boldsymbol{r}_{il_i}-max_{j\neq l_i}\boldsymbol{r}_{ij}\ge C,\end{aligned}\tag{8}$$

where $\boldsymbol{A}\in\Re^{d\times s}$ and $\boldsymbol{B}\in\Re^{s\times c}$. It is easy to find that Eqn. (8) has the same solution of Eqn. (7), but $\boldsymbol{W}$ has more interpretable yet discriminative low-rank property. Interestingly, Eqn. (8) can be written as

$$\begin{aligned}&\min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{r}_i}\sum_{i=1}^{n}\|\boldsymbol{B}^T(\boldsymbol{A}^T\boldsymbol{x}_i)-\boldsymbol{r}_i\|_2^2\\&s.t.\ \boldsymbol{r}_{il_i}-max_{j\neq l_i}\boldsymbol{r}_{ij}\ge C.\end{aligned}\tag{9}$$

It is worth noting that matrix $\boldsymbol{A}$ can be viewed as a projection that transforms the data from the original feature space to a latent analytic subspace. Specifically, for each data point $\boldsymbol{x}_i\in\Re^d$, its corresponding representation in the latent subspace should be $\boldsymbol{A}^T\boldsymbol{x}_i\in\Re^s$.

To further explore the underlying predictive structures, we assume that the discriminative mapping consists of two components: one is the observed high-dimensional feature map, and the other one is the latent low-dimensional feature map. In other words, the preferable linear predictor has the form as

$$f_k(x_i)=\boldsymbol{p}_k^T\boldsymbol{x}_i+\boldsymbol{b}_k^T(\boldsymbol{A}^T\boldsymbol{x}_i)\tag{10}$$

where $\boldsymbol{p}_k\in\Re^d$ and $\boldsymbol{b}_k\in\Re^s$ herein can be viewed as the weighting vectors for each specific predictor. Instead of only using the latent prediction in Eqn. (9), the linear structural prediction is constructed as

$$\begin{aligned}&\min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{r}_i}\sum_{i=1}^{n}\|(\boldsymbol{P}+\boldsymbol{A}\boldsymbol{B})^T\boldsymbol{x}_i-\boldsymbol{r}_i\|_2^2\\&s.t.\ \boldsymbol{r}_{il_i}-max_{j\neq l_i}\boldsymbol{r}_{ij}\ge C.\end{aligned}\tag{11}$$

where $\boldsymbol{P}=[\boldsymbol{p}_1,\cdots,\boldsymbol{p}_c]$ and $\boldsymbol{B}=[\boldsymbol{b}_1,\cdots,\boldsymbol{b}_c]$. To simplify the optimization problem, we define $\boldsymbol{W}=\boldsymbol{P}+\boldsymbol{A}\boldsymbol{B}$. Therefore, we may eliminate $\boldsymbol{P}$ by using $\boldsymbol{W}$, $i.e.$ $\|\boldsymbol{P}\|_F^2=\|\boldsymbol{W}-\boldsymbol{A}\boldsymbol{B}\|_F^2$, and the resulting predictive functions learning is formulated as

$$\begin{aligned}&\min_{\boldsymbol{W},\boldsymbol{A},\boldsymbol{B},\boldsymbol{r}_i}\sum_{i=1}^{n}\|\boldsymbol{W}^T\boldsymbol{x}_i-\boldsymbol{r}_i\|_2^2+\gamma\|\boldsymbol{W}-\boldsymbol{A}\boldsymbol{B}\|_F^2\\&s.t.\ \boldsymbol{r}_{il_i}-max_{j\neq l_i}\boldsymbol{r}_{ij}\ge C.\end{aligned}\tag{12}$$

To make the problem tractable, we add an orthogonal constraint on $\boldsymbol{A}$, $i.e.$ $\boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I}$. In addition, to obtain a more stable result of problem (11), the Frobenius norm of $\boldsymbol{W}$ is employed to capture the group characteristics of data [27], [29], and then we define the following regularization term to control the complexity of the optimization problem:

$$\Psi(f)=\|\boldsymbol{W}\|_F^2+\frac{\gamma}{\beta}\|\boldsymbol{W}-\boldsymbol{A}\boldsymbol{B}\|_F^2\ s.t.\ \boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I}.\tag{13}$$

C. Adaptive Graph Structure Learning

It is easy to see that learning discriminative regression targets is to highlight the inter-class separation of the learned representation, but the intra-class compactness is significant to discriminative representation learning as well. We propose to construct an adaptive probabilistic graph to improve the compactness of the learned representation. Specifically, it is known that the pairwise similarity reflects the probabilistic connectivity between each pair of samples. The conventional unsupervised methods such as LLE [14] and LPP [15], construct data connectivity based on distance similarity on the original space, which contains many possible redundant and noise information. So, these data connectivity can not faithfully capture the data structure and similarity, and using such similarity matrices are surely unreliable and inaccurate for the regression task. To this end, our paper applies an adaptive process to determine the similarity matrix by fully considering both supervised semantic label information and unsupervised distance information. A probabilistic graph learning model is built in the discriminative projected semantic space by assigning the optimal neighbors for each data point based on the local distances. It assumes that the closely-related predicted targets should have higher possibilities to be connected, $i.e.$

$$\begin{aligned}\Phi(f)=\sum_{i,j=1}^{n}dist(\boldsymbol{W}^T\boldsymbol{x}_i,\boldsymbol{W}^T\boldsymbol{x}_j)\times\boldsymbol{P}_{ij}\\s.t.\ 0<\boldsymbol{P}_{ij}<1,\boldsymbol{P}\boldsymbol{e}_n=\boldsymbol{e}_n,\end{aligned}\tag{14}$$

where $dist(a,b)$ measures the distance between a and b. The constraints guarantee $\boldsymbol{P}$ being a transition probability matrix, $i.e.$ each of its rows is a probability distribution. In this work, we simply define the distance between two predicted targets as the square of Euclidean distance. Furthermore, instead of using a fixed probability matrix $\boldsymbol{P}$, we use a self-tuning technique to learn a more feasible similarity measurement adaptively. To achieve this goal and avoid a trivial solution of $\boldsymbol{P}$, we add a simple constraint on $\boldsymbol{P}$ as

$$\begin{aligned}\Phi(f)=\sum_{i,j=1}^{n}\left(\|\boldsymbol{W}^T\boldsymbol{x}_i-\boldsymbol{W}^T\boldsymbol{x}_j\|_2^2\boldsymbol{P}_{ij}+\sigma\boldsymbol{P}_{ij}^2\right)\\s.t.\ 0<\boldsymbol{P}_{ij}<1,\boldsymbol{P}\boldsymbol{e}_n=\boldsymbol{e}_n,\end{aligned}\tag{15}$$

where σ is a nonnegative trade-off parameter, but it can be automatically determined (shown in Section IV-C). It is notable that the adaptive graph learning exploits the probabilistic data connectivity in the learned discriminative projected space to enhance the intra-class compactness.

Therefore, by combining the loss function (6), the complexity regularization (12) and the graph smoothness term (15), the final formulation of the proposed MSRL is

$$\begin{aligned}&\Gamma(\boldsymbol{W},\boldsymbol{A},\boldsymbol{B},\boldsymbol{R},\boldsymbol{P})=\|\boldsymbol{W}^T\boldsymbol{X}-\boldsymbol{R}\|_F^2+\gamma\|\boldsymbol{W}-\boldsymbol{A}\boldsymbol{B}\|_F^2\\&\quad+\beta\|\boldsymbol{W}\|_F^2+\lambda\sum_{i,j=1}^{n}\left(\|\boldsymbol{W}^T\boldsymbol{x}_i-\boldsymbol{W}^T\boldsymbol{x}_j\|_2^2\boldsymbol{P}_{ij}+\sigma\boldsymbol{P}_{ij}^2\right)\\&s.t.\ \boldsymbol{R}_{il_i}-\max_{j\neq l_i}\boldsymbol{R}_{ij}\ge C,\boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I},\\&\quad0\le\boldsymbol{P}_{ij}\le1,\boldsymbol{P}\boldsymbol{e}_n=\boldsymbol{e}_n.\end{aligned}\tag{16}$$

From the above objective function, it is easy to see that the learned data representations $\boldsymbol{W}^T\boldsymbol{X} \in \Re^{c\times n}$ naturally own the following attributes.

1) The learned representations are globally consistent. By optimizing the regression targets with a marginal fine-tuning constraint, the learned representations are consistent with the groundtruth labels but discriminative.
2) The learned data representations are locally consistent. Specifically, the adaptive similarity matrix $\boldsymbol{P}$ assigns the adaptive and optimal neighbors for each data point based on the probabilistic connectivity. The learned representations are consistent with the nearby points in the discriminative projected target space.
3) The forth term can avoid the over-fitting problem induced by the effective context correlations $\boldsymbol{W}$, and also interpolates the semantic links in designing the probabilistic graph model with self-adaptation.
4) By jointly optimizing the mapping function and learning marginal representations, it is helpful to mutually guide its counterpart to achieve more favorable results. The proposed framework ensures that the learned data representations are discriminative enough.
5) Unlike existing algorithms, the proposed method is more flexible in that (a) it does not use the strict binary regression targets but learns them from data by enlarging the margins of different classes for each sample and (b) the weights of graph embedding are automatically inferred from the projected data instead of a fixed setting.

IV. Solving the Optimization Problem

For the optimization problem (16), it is easy to verify that the objective function cannot be directly optimized because the variables ($i.e.$ $\boldsymbol{W}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{R}, \boldsymbol{P}$) depend on each other. Consequently, an iterative optimization algorithm is developed.

A. Update W, A, B given R and P

Following the common optimization process, the block coordinate descent method is employed by iteratively updating $\boldsymbol{W}, \boldsymbol{A}, \boldsymbol{B}$ with fixed $\boldsymbol{R}$ and $\boldsymbol{P}$, and then the optimization problem (16) can be written as

$$\begin{aligned}\Gamma(\boldsymbol{W},\boldsymbol{A},\boldsymbol{B}) &= \|\boldsymbol{W}^T\boldsymbol{X}-\boldsymbol{R}\|_F^2+\gamma\|\boldsymbol{W}-\boldsymbol{A}\boldsymbol{B}\|_F^2+\beta\|\boldsymbol{W}\|_F^2 \\ &+\lambda\sum_{i,j=1}^{n}\|\boldsymbol{W}^T\boldsymbol{x}_i-\boldsymbol{W}^T\boldsymbol{x}_j\|_2^2\boldsymbol{P}_{ij}\ s.t.\ \boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I}, \end{aligned} \tag{17}$$

which can be converted into an equivalent problem:

$$\begin{aligned}\Gamma(\boldsymbol{W},\boldsymbol{A},\boldsymbol{B}) &= \|\boldsymbol{W}^T\boldsymbol{X}-\boldsymbol{R}\|_F^2+\gamma\|\boldsymbol{W}-\boldsymbol{A}\boldsymbol{B}\|_F^2+\beta\|\boldsymbol{W}\|_F^2 \\ &+\lambda tr(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{W})\ s.t.\ \boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I}, \end{aligned} \tag{18}$$

where $\boldsymbol{L}$ is the graph Laplacian matrix of $\boldsymbol{P}$, which is defined as $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{P}$, and $\boldsymbol{D}$ is a diagonal matrix whose main diagonal elements are column sums of matrix $\boldsymbol{P}$, that is, $\boldsymbol{D}_{ii}=\sum_{j=1}^{n}\boldsymbol{P}_{ij}$.

Updating $\boldsymbol{B}$: It is easy to find that the optimal $\boldsymbol{B}$ in Eqn. (18) can be expressed in terms of $\boldsymbol{A}$ and $\boldsymbol{W}$. Based on constraint $\boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I}$, we can infer the solution of $\boldsymbol{B}$ by setting the first-order derivation $\frac{\partial\Gamma}{\partial\boldsymbol{B}}=\boldsymbol{0}$, and then

$$2\gamma(\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{B}-\boldsymbol{A}^T\boldsymbol{W})=0 \ \Leftrightarrow\ \boldsymbol{B}=\boldsymbol{A}^T\boldsymbol{W}. \tag{19}$$

Updating $\boldsymbol{W}$: Similarly, when we fix $\boldsymbol{A}$ and $\boldsymbol{B}$ and substitute $\boldsymbol{B}$ in Γ with Eqn. (18), objective function Γ is reformulated as follows:

$$\begin{aligned}\Gamma =&\|\boldsymbol{W}^T\boldsymbol{X}-\boldsymbol{R}\|_F^2+\lambda tr(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{W}) \\ &+\beta\|\boldsymbol{W}\|_F^2+\gamma tr[\boldsymbol{W}^T(\boldsymbol{I}-\boldsymbol{A}\boldsymbol{A}^T)(\boldsymbol{I}-\boldsymbol{A}\boldsymbol{A}^T)\boldsymbol{W}]. \end{aligned} \tag{20}$$

As $(\boldsymbol{I}-\boldsymbol{A}\boldsymbol{A}^T)(\boldsymbol{I}-\boldsymbol{A}\boldsymbol{A}^T)=\boldsymbol{I}-\boldsymbol{A}\boldsymbol{A}^T$, by setting the first-order derivation $\frac{\partial\Gamma}{\partial\boldsymbol{W}}=\boldsymbol{0}$ for Eqn. (20), it leads to

$$\boldsymbol{W}=(\boldsymbol{G}-\gamma\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{X}\boldsymbol{R}^T, \tag{21}$$

where $\boldsymbol{G}=\boldsymbol{X}\boldsymbol{X}^T+\lambda\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T+(\beta+\gamma)\boldsymbol{I}$.

Updating $\boldsymbol{A}$: To obtain a more efficient solution, we give a simple solution. By ignoring the constant terms independent of $\boldsymbol{A}$, minimizing (18) becomes:

$$\min_{\boldsymbol{A}}\|\boldsymbol{W}-\boldsymbol{A}\boldsymbol{B}\|_F^2\ s.t.\ \boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I}, \tag{22}$$

which is the famous Orthogonal Procrustes problem. The following lemma gives the optimal solution to the optimization problem (22).

Lemma 1: *Let the SVD of* $\boldsymbol{W}\boldsymbol{B}^T=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, *and then* $\boldsymbol{A}=\boldsymbol{U}\boldsymbol{V}^T$ *is the optimal solution of problem (22).*

B. Update R given P, W, A, B

Similarly, if we remove the constant terms independent of $\boldsymbol{R}$, the objective function Γ with Eqn. (16) is written as follows:

$$\min_{\boldsymbol{R}}\|\boldsymbol{W}^T\boldsymbol{X}-\boldsymbol{R}\|_F^2\ s.t.\ \boldsymbol{R}_{il_i}-\max_{j\neq l_i}\boldsymbol{R}_{ij}\geq C. \tag{23}$$

Similar to SVM [32], we set the marginal value of constant $C=1$ here, and then problem (23) is reformulated as

$$\min_{\boldsymbol{R}}\|\boldsymbol{F}-\boldsymbol{R}\|_F^2\ s.t.\ \boldsymbol{R}_{il_i}-\max_{j\neq l_i}\boldsymbol{R}_{ij}\geq 1, \tag{24}$$

where $\boldsymbol{F}=\boldsymbol{W}^T\boldsymbol{X}$. Problem (24) is a convex constrained quadratic programming problem [37] and we decompose it into n independent subproblems. For the i-th columns of $\boldsymbol{F}$ and $\boldsymbol{R}$, we denote $\boldsymbol{f}=[\boldsymbol{f}_1,\cdots,\boldsymbol{f}_c]^T\in\Re^c$ and $\boldsymbol{r}=[\boldsymbol{r}_1,\cdots,\boldsymbol{r}_c]^T\in\Re^c$. Assuming that the i-th sample is from the mth-class, the i-th subproblem of problem (24) is

$$\min_{\boldsymbol{r}}\|\boldsymbol{f}-\boldsymbol{r}\|_2^2=\sum_{j=1}^{c}(\boldsymbol{f}_j-\boldsymbol{r}_j)_2^2\ s.t.\ \boldsymbol{r}_m-\max_{j\neq m}\boldsymbol{r}_j\geq 1. \tag{25}$$

To optimize problem (25), we introduce an auxiliary variable $\boldsymbol{z}\in\Re^c$, and $\boldsymbol{z}_j=\boldsymbol{f}_j+1-\boldsymbol{f}_m$, where $\boldsymbol{z}_j\leq 0$ indicates the predictive targets are coincident with the marginal constraint, otherwise unsatisfied results. We assume that $\boldsymbol{r}_m=\boldsymbol{f}_m+\zeta$, where ζ is a learning factor. For $\forall j\neq m$, $\boldsymbol{r}_m-\boldsymbol{r}_j\geq 1$, and then the j-th subproblem of (25) is

$$\min_{\boldsymbol{r}_j}(\boldsymbol{f}_j-\boldsymbol{r}_j)_2^2\ s.t.\ \boldsymbol{f}_m+\zeta-\boldsymbol{r}_j\geq 1,\forall j\neq m. \tag{26}$$

Algorithm 1. Solving Problem (25)

Input: $\boldsymbol{r} = [\boldsymbol{r}_1, \cdots, \boldsymbol{r}_c]^T \in \Re^c$, the true class index m.
Initialization: $\forall j, \boldsymbol{z}_j = \boldsymbol{f}_j + 1 - \boldsymbol{f}_m$, $\zeta = 0$, $t = 0$.
for $j \neq m$ **do**
  if $\psi'(\boldsymbol{z}_j) > 0$ **then** $\zeta = \zeta + \boldsymbol{z}_j, t = t + 1$ **end**
end
Define $\zeta = \zeta/(1 + t)$, and then update $\boldsymbol{r}_j$ by Eqn.(27).
Output: Marginal target vector $\boldsymbol{r}$.

Clearly, the above problem is a simple quadratic programming problem, and the optimal solution is $\boldsymbol{r}_j = \boldsymbol{f}_j + min(\zeta - \boldsymbol{z}_j, 0)$ and then the optimal solution of problem (25) is given by

$$\boldsymbol{r}_j = \begin{cases} \boldsymbol{f}_j + \zeta, & if\ j = m, \\ \boldsymbol{f}_j + min(\zeta - \boldsymbol{z}_j, 0), & otherwise. \end{cases} \tag{27}$$

Thus, problem (25) is transformed into

$$\min_{\zeta} \psi(\zeta) = \zeta^2 + \sum_{j \neq m} (min(\zeta - \boldsymbol{z}_j, 0))^2, \tag{28}$$

where its first-order derivation $\psi'(\zeta) = 2(\zeta + \sum_{j \neq m} min(\zeta - \boldsymbol{z}_j, 0))$. By taking $\psi'(\zeta) = 0$, the optimal value of learning factor ζ is calculated as

$$\zeta = \frac{\sum_{j \neq m} \boldsymbol{z}_j \Omega(\psi'(\boldsymbol{z}_j) > 0)}{1 + \sum_{j \neq m} \boldsymbol{z}_j \Omega(\psi'(\boldsymbol{z}_j) > 0)}, \tag{29}$$

where $\Omega(\cdot)$ is the indicator operator. The detailed procedures of learning the optimal solution of the i-th column of $\boldsymbol{R}$ are summarized in Algorithm 1.

C. Update P given R, W, A, B

By removing all the irrelevant terms with respect to $\boldsymbol{P}$, the objective function of problem (16) is written as follows:

$$\begin{aligned} &\sum_{i,j=1}^{n} \left(\|\boldsymbol{f}_i - \boldsymbol{f}_j\|_2^2 \boldsymbol{P}_{ij} + \sigma \boldsymbol{P}_{ij}^2 \right) \\ &s.t.\ 0 \leq \boldsymbol{P}_{ij} \leq 1, \boldsymbol{P}\boldsymbol{e}_n = \boldsymbol{e}_n, \end{aligned} \tag{30}$$

where $\boldsymbol{f}_i = \boldsymbol{W}^T \boldsymbol{x}_i$, and it can be decoupled into n independent subproblems, and each of them has the following form

$$\begin{aligned} &\min_{\boldsymbol{p}_i} \sum_{j=1}^{n} \left(\|\boldsymbol{f}_i - \boldsymbol{f}_j\|_2^2 \boldsymbol{P}_{ij} + \sigma \boldsymbol{P}_{ij}^2 \right) \\ &s.t.\ 0 \preceq \boldsymbol{p}_i \preceq 1, \boldsymbol{p}_i^T \boldsymbol{e}_n = 1, \end{aligned} \tag{31}$$

By defining $\boldsymbol{d}_{ij} = -\frac{1}{2\sigma} \|\boldsymbol{f}_i - \boldsymbol{f}_j\|_2^2$, problem (31) can be further written as follows

$$\min_{\boldsymbol{p}_i^T} \frac{1}{2} \|\boldsymbol{d}_i^T - \boldsymbol{p}_i^T\|_2^2 \ s.t.\ 0 \preceq \boldsymbol{p}_i^T \preceq 1, \boldsymbol{p}_i^T \boldsymbol{e}_n = 1, \tag{32}$$

where $\boldsymbol{p}_i$ and $\boldsymbol{d}_i$ here are the i-th rows of $\boldsymbol{P}$ and $\boldsymbol{D}$, respectively. Its Lagrangian function is

$$\min_{\boldsymbol{p}_i^T} \frac{1}{2} \|\boldsymbol{d}_i^T - \boldsymbol{p}_i^T\|_2^2 - \mu(\boldsymbol{p}_i^T \boldsymbol{e}_n - 1) - \eta \boldsymbol{p}_i^T, \tag{33}$$

where μ and η are the Lagrangian multipliers. To achieve better performance and accelerate the computation time, it is favorable to learn a k-sparse $\boldsymbol{p}_i$, $i.e.$ only the k-nearest neighbors are preserved to be locally connected. Based on the KKT condition, problem (33) has a closed form solution:

$$\boldsymbol{p}_i = max(\boldsymbol{d}_i + z, 0),\ z = \frac{1}{k}(1 - \sum_{j=1}^{k} \breve{\boldsymbol{d}}_{ij}), \tag{34}$$

where $\breve{\boldsymbol{d}}_i$ is a sorted vector of $\boldsymbol{d}_i$ in an ascending order. Moreover, for each subproblem we have the following inequality

$$\frac{k}{2} \boldsymbol{d}_{i,k} - \frac{1}{2} \sum_{j=1}^{k} \boldsymbol{d}_{i,j} < \sigma_i \leq \frac{k}{2} \boldsymbol{d}_{i,k+1} - \frac{1}{2} \sum_{j=1}^{k} \boldsymbol{d}_{i,j}. \tag{35}$$

Because we expect that only k nearest neighbors are used to construct the local information of the data, $i.e.$ exactly k non-zero values in σ_i, σ_i has to satisfy the above inequality, and the mean value of non-zero entries in each row of $\boldsymbol{P}$ is approximate to k ($k = 15$ in this work). Therefore, the value of σ could be set to

$$\sigma = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{k}{2} \boldsymbol{d}_{i,k+1} - \frac{1}{2} \sum_{j=1}^{k} \boldsymbol{d}_{i,j} \right). \tag{36}$$

Based on the above analysis, an iterative optimization method is developed to solve the objective function of (16), and the main optimization steps of MSRL are summarized in Algorithm 2. The convergence criterion used in our experiments is that the number of iterations is up to 30 or $|\Gamma_{t+1} - \Gamma_t|/\Gamma_t < 0.001$, where Γ_t is the value of the objective function in the t-th iteration. Once the regression matrix $\boldsymbol{W}$ is obtained, we directly use $\boldsymbol{W}$ to obtain the learned data representations of training and test samples, respectively. Finally, we employ the simple nearest-neighbor (NN) classifier to make final recognition.

D. Semi-supervised Extension of MSRL

In this subsection, we show that the proposed MSRL can be easily extended to its semi-supervised case. There is a set of labeled data with l instances $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_l]$, and the unlabeled data with u instances $\hat{\boldsymbol{X}} = [\hat{\boldsymbol{x}}_1, \cdots, \hat{\boldsymbol{x}}_u]$. All the available data are denoted as $\tilde{\boldsymbol{X}} = [\boldsymbol{X}, \hat{\boldsymbol{X}}] \in \Re^{d \times N}$ and $N = l + u$. It is notable that the labels of $\boldsymbol{X}$ are used to train the marginal target $\boldsymbol{R}$. The resulting semi-supervised MSRL (SMSRL) can be developed as:

$$\begin{aligned} &\min_{\boldsymbol{W}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{R}, \boldsymbol{P}} \|\boldsymbol{W}^T \boldsymbol{X} - \boldsymbol{R}\|_F^2 + \gamma \|\boldsymbol{W} - \boldsymbol{A}\boldsymbol{B}\|_F^2 + \beta \|\boldsymbol{W}\|_F^2 \\ &\quad + \lambda \sum_{i,j=1}^{N} \left(\|\boldsymbol{W}^T \tilde{\boldsymbol{x}}_i - \boldsymbol{W}^T \tilde{\boldsymbol{x}}_j\|_2^2 \boldsymbol{P}_{ij} + \sigma \boldsymbol{P}_{ij}^2 \right) \\ &s.t.\ \boldsymbol{R}_{il_i} - \max_{j \neq l_i} \boldsymbol{R}_{ij} \geq C, \boldsymbol{A}^T \boldsymbol{A} = \boldsymbol{I}, \\ &\quad 0 \leq \boldsymbol{P}_{ij} \leq 1, \boldsymbol{P}\boldsymbol{e}_n = \boldsymbol{e}_n, \end{aligned} \tag{37}$$

where $\boldsymbol{P} \in \Re^{N \times N}$, and $\tilde{\boldsymbol{x}}_i$ denotes the i-th sample from the whole dataset, *i.e.* $\tilde{\boldsymbol{X}}$. Specifically, the first term is devoted to learn discriminative mapping $\boldsymbol{W}$ from the labeled data $\boldsymbol{X}$ to the marginalized targets $\boldsymbol{R}$, while the second and third terms are the same as MSRL for structural predictor learning. The last term is designed to construct adaptive probabilistic graph

Algorithm 2. Solving the MSRL Optimization Problem

Input: Feature Matrix $\boldsymbol{X}$; Label Matrix $\boldsymbol{Y}$;
Parameters $\lambda, \beta, \gamma, k$ and s.
1: Set $iter = 0$; Initialize $\boldsymbol{W}$ using basic LSR, $\boldsymbol{R} = \boldsymbol{Y}$, and set $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}$ as identity matrices;
2: **repeat**
3: $\boldsymbol{B} = \boldsymbol{A}^T\boldsymbol{W}$;
4: $\boldsymbol{G} = \boldsymbol{X}\boldsymbol{X}^T + \lambda\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T + (\beta+\gamma)\boldsymbol{I}$;
5: $\boldsymbol{W} = (\boldsymbol{G} - \gamma\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{X}\boldsymbol{R}^T$;
6: $[\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}^T] = svd(\boldsymbol{W}\boldsymbol{B}^T)$;
7: $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{V}^T$;
8: Update the target matrix $\boldsymbol{R}$ column-by-column using Algorithm 1;
9: Update the i-th row of $\boldsymbol{P}$, $i.e.$ $\boldsymbol{p}_i$, row-by-row using Eqn.(34);
10: $iter = iter + 1$;
11: **until** Convergence criterion satisfied
Output: Converged $\boldsymbol{W}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{T}, \boldsymbol{P}$.

on both labeled and unlabeled data. It is notable that *the semantic labels of the labeled samples* are employed to optimize the marginalized targets $\boldsymbol{R}$. SMSRL employs the adaptive graph regularization term to build the relations between the labeled and unlabeled data, which is a common strategy to build the semi-supervised model. Specifically, SMSRL takes full consideration of all the available samples $\tilde{\boldsymbol{X}}$ based on the graph regularization. The optimization problem (37) can be solved by Algorithm 2, where the graph Laplacian matrix is constructed on all the available samples $\tilde{\boldsymbol{X}}$, and $\boldsymbol{R}$ is learned on the labeled samples $\boldsymbol{X}$.

V. Algorithm Analysis

A. Convergence Analysis

As shown in Algorithms 1 and 2, we can easily get the following proposition.

Proposition 1: *The optimization problem (16) is convex with respect to* $\boldsymbol{W}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{R}$ *and* $\boldsymbol{P}$*, respectively.*

Proof. The detailed proof of Proposition 1 is moved to Appendix A for better flow of the paper. □

Based on Proposition 1, the proposed iterative algorithm can be demonstrated to converge to a unique optimal solution, and the following theorem is satisfied.

Theorem 1: *The iterative optimization algorithm shown in Section IV monotonically decreases the value of the objective function (16) in each iteration.*

Proof. The detailed proof of Theorem 1 is moved to Appendix B for better flow of the paper. □

Based on Theorem 1, it is easy to see that the developed iterative method in Algorithm 2 can converge to a local optimal solution. In the experiments, we demonstrate that our algorithm can efficiently converge within 30 iterations.

B. Computational Complexity Analysis

In this section, we briefly analyze the computation complexity of the proposed iterative optimization method for the MSRL model. In each iteration, the cost of calculating $\boldsymbol{B}$ needs $\mathcal{O}(dns)$, where d, n and s are the dimension of features, the number of labeled images and the dimension of the latent subspace, respectively. The complexity of computing $\boldsymbol{G}$ is $\mathcal{O}(dn^2 + d^2n)$. We notice that the construction of $\boldsymbol{W}$ is computed with the cost of $\mathcal{O}(dn^2 + d^2n + dnc)$, while the complexity of obtaining $\boldsymbol{A}$ is $\mathcal{O}(d^2s + ds^2)$. As shown in Algorithm 1, $\boldsymbol{R}$ is computed with the cost of $\mathcal{O}(nc)$. Finally, the probabilistic matrix $\boldsymbol{P}$ is obtained with the time complexity of $\mathcal{O}(cn^2)$. Thus, the total cost of our MSRL method for each iteration is $\mathcal{O}(dnc + dn^2 + d^2n)$ on account of $s < d$ and $s < n$ in our experiments.

It is worth pointing out that our MSRL can converge within 30 iterations, which will be testified in the experiment section. So, the computational cost of our method is acceptable. By comparison, we briefly analyze the computational complexities of some compared methods as follows. For SRC [5], it should iteratively optimize (N-n) independent l_1-norm regularization problems [5], [7], [10], and its computational complexity is about $\mathcal{O}((N-n)(n^2+nd)$, which is slower than our method due to more iterations. RPCA and LRSI include two phases, i.e. learning representations and SRC based classification, which are much slower than our method. For some accelerated methods such as SLRM, LLC, LRC and CRC, they need to compute the representation coefficients of the training samples, and then calculate the representation residuals of each class for classification, which have similar computational costs of our method. The computation complexities of LRLR, LRRR and SLRR are about $\mathcal{O}(dn + n^2d)$, which is little faster than our method. For RLSL, DKSVD and LC-KSVD, they have similar or little higher computation complexity of our method in a single iteration, but the number of iterations are much larger than ours. The low-rank and sparse representation based methods such as SRRS, CBDS and LatLRR need at least $\mathcal{O}(n^3 + n^2d)$, because they simultaneously computes SVD of feature matrix and solves simple sparse optimization problem. In generally, the overall computation burden of our MSRL is lower than these low-rank and sparse representation methods.

C. Connection to some previous algorithms

In this subsection, we establish the relationships between our method and some related data representation learning algorithms, including LSR, DLSR [29], SVM [32], LRRR [24], and the semi-supervised low-rank mapping (SLRM) method [34].

1) Comparison of MSRL, LSR and DLSR: From the objective function of LSR in (1), we can see that LSR aims at learning a regression matrix $\boldsymbol{W}$ and enforces the regression results to approximate a zero-one matrix, $i.e.$ 0 and 1 respectively represent the false and true classes. On the other hand, DLSR using (2) projects the original features to a positive-negative matrix, that is, the positive and negative values denote the true and false classes, respectively. However, both LSR and DLSR utilize the conventional zero-one matrix as their regression targets, which can not precisely estimate the regression results. In contrast, our MSRL method directly learns the regression targets for each data point, and enforces the regression results to have relative marginal characteristics. That is, the values of the true classes are larger than that of the false classes with a certain criterion. Therefore, MSRL not only guarantees the flexible but marginal regression targets but also can more

accurately measure the regression results in comparison with LSR and DLSR.

Moreover, other important advantages are that MSRL uses the discriminative features in the discriminant latent subspace to predict the regression targets instead of the original feature. The adaptive graphical structure is employed to avoid the overfitting deficiency. As a result, MSRL is essentially a favorable and discriminative learning method for data representation.

2) Comparison of MSRL and Hinge Loss in SVM: The hinge loss in SVM is closely related but different to the marginal regression targets learning process. Based on the variables' definitions in (24) and (25), the hinge loss is

$$\delta_i = \max\{1 + \max_{j \neq m} \boldsymbol{F}_{ij} - \boldsymbol{F}_{im}, 0\}, \tag{38}$$

The classifiers are formulated as

$$\min_{\boldsymbol{W}} \sum_{i=1}^{n} \delta_i + \alpha \|\boldsymbol{W}\|_F^2, \ or \ \min_{\boldsymbol{W}} \sum_{i=1}^{n} \delta_i^2 + \alpha \|\boldsymbol{W}\|_F^2, \tag{39}$$

which are l_1-SVM and l_2-SVM, respectively. It is observed that the above functions are the modification of the empirical loss functions other than redefinition of target matrix of LSR. Therefore, the discriminative target learning in MSRL can be considered as a reformulation of the hinge loss for data representation. Moreover, MSRL is better than SVM. The main reason is that, in addition to marginalized regression target learning, MSRL optimizes the projection matrix $\boldsymbol{W}$ by exploring the linear structural predictor learning and improves the compactness of the projected data with the adaptive probabilistic graph regularization.

3) Comparison of SMSRL and SLRM: Since SLRM is a semi-supervised method, we compare our SMSRL with SLRM. SLRM identifies the favorable mapping function by collaboratively exploiting the low-rank constraint to capture the correlations between labels and constructing a manifold regularization to preserve the geometric structure of data. The objective function of SLRM is

$$\|\boldsymbol{W}^T \tilde{\boldsymbol{X}} - \boldsymbol{Y}\|_F^2 + \beta \|\boldsymbol{W}\|_* + \lambda tr(\|\boldsymbol{W}^T \boldsymbol{X} \bar{\boldsymbol{L}} \boldsymbol{X}^T \boldsymbol{W}), \tag{40}$$

where $\boldsymbol{P}_{ij} = \exp\left(\frac{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\varrho^2}\right)$ is the similarity matrix defined in their paper, and $\bar{\boldsymbol{L}}$ is defined similarly to (18). $\|\boldsymbol{W}\|_*$ is the nuclear norm of $\boldsymbol{W}$, which is a convex relaxation of $low\text{-}rank$ minimization. Based on the observation [24]: $\|\boldsymbol{W}\|_* = \min_{\boldsymbol{W}=\boldsymbol{AB}} \frac{1}{2}(\|\boldsymbol{A}\|_F^2 + \|\boldsymbol{B}\|_F^2)$, the proposed MSRL problem (37) can be reformulated as

$$\begin{aligned} \Gamma = &\|\boldsymbol{W}^T \tilde{\boldsymbol{X}} - \boldsymbol{R}\|_F^2 + \gamma \|\boldsymbol{W}\|_* + \beta \|\boldsymbol{W}\|_F^2 \\ &+ \lambda(tr(\|\boldsymbol{W}^T \boldsymbol{X} \boldsymbol{L} \boldsymbol{X}^T \boldsymbol{W}) + \sigma \|\boldsymbol{P}\|_F^2) \ s.t. \ \boldsymbol{A}^T \boldsymbol{A} = \boldsymbol{I}, \\ &\boldsymbol{R}_{il_i} - \max_{j \neq l_i} \boldsymbol{R}_{ij} \geq C, 0 \leq \boldsymbol{P}_{ij} \leq 1, \boldsymbol{P}\boldsymbol{e}_n = \boldsymbol{e}_n. \end{aligned} \tag{41}$$

It is easy to see that the distance metric $\boldsymbol{Q}$ is fixed prior without any adaptation, while $\boldsymbol{P}$ in SMSRL is an automatically self-tuning metric in each iteration. Moreover, without considering the adaptive graph-embedding regularization, the first three terms in (41) have the following proposition.

Proposition 2: *The first three terms of the objective function (41) lead to a discriminative linear regression problem with an elastic-net regularization of singular values.*

Proof. The detailed proof of Proposition 2 is moved to Appendix C for better flow of the paper. □

It is known that the elastic-net regularization is a robust model. Therefore, SLRM is a special case of SMSRL, while SMSRL is more discriminative than SLRM.

VI. Experimental Results

To evaluate the proposed MSRL framework, we compare it with 23 state-of-the-art data representation learning methods. We empirically validate the superiority of our method on four different but related applications, including object, face, texture and scene recognition. The promising experimental results show that our MSRL and SMSRL are better than some representative data representation learning algorithms. Furthermore, extensive experimental analyses demonstrate that our method is of a well-balanced tradeoff between the discriminative capability, efficiency and effectiveness.

A. Experimental Settings

We conduct comparative experiments with the state-of-the-art data representation learning algorithms, including low-rank linear regression (LRLR) [24], LRRR [24], sparse low-rank regression (SLRR) [24], SVM [32], capped SVM (CapSVM) [33], CRC [18], LRC [20], LLC [19], SRC [5], ProCRC [38], CBDS [27], DLSI [26], discriminative sparse representation method (DSRM) [6], RLSL [39], DLSR [29], RPCA [21], LatLRR [22], SLRM [34], DKSVD [12], LC_KSVD [13], and manifold-inspired representation learning algorithm, $i.e.$ principal coefficients embedding (PCE) [16] and LDA [4]. It should be noted that SLRM [34] is a semi-supervised method. To show the indispensability of the adaptive graph learning, we remove the adaptive graph term, and denote the remaining part as MSRL-G. All the algorithms are repeated ten times with different random splits of training and test data.

To make fair comparisons, we re-implemented all the compared methods by using the released codes from the corresponding authors. We search the best parameters for each algorithm by tenfold cross-validation, or directly use the suggested parameter settings. Specifically, LatLRR is used for feature extraction, and then we utilize the obtained latent features to fit model (1) followed by the 1-NN classifier. For RPCA, we first use RPCA to preprocess the samples, and then employ SRC for classification [26]. For LLC, the numbers of local bases for LLC and LLC* are set to 15 and 30 respectively, which are similar to [13], [19]. In addition, the one-versus-rest rule is exploited in SVM to learn training parameters, and the LibSVM [32] toolbox is employed for recognition. The important cost parameter C in SVM is selected by cross validation from the candidate set $\{0.001, 0.01, 0.1, 1.0, 10.0\}$. In our experiments, the values of γ in both MSRL and SMSRL are simply fixed with 0.05. To fairly compare different algorithms, the optimal parameters are cross-validated from the candidate set $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0\}$ for achieving optimal recognition accuracies. *In order to obtain better recognition results, in the training phase of SMSRL, all the training samples used for MSRL are treated as the labeled samples of SMSRL, and the test samples are used*

TABLE I: Recognition accuracies (mean±std %) of different methods on the PFID database.

Alg.	6	8	10	12
LRLR	41.13±1.18	47.03±2.42	49.41±1.96	52.81±2.62
LRRR	49.62±1.91	52.90±1.12	53.18±1.44	55.49±2.29
SLRR	49.44±2.07	52.95±1.08	54.36±1.40	56.09±1.85
SVM	52.02±1.31	57.44±1.61	60.51±1.69	63.03±2.05
CapSVM	55.71±1.71	60.57±2.54	64.63±0.92	66.83±1.12
CRC	52.54±1.74	55.54±1.71	57.34±1.01	59.54±2.71
LLC	54.88±1.99	57.98±1.70	59.41±2.35	62.08±2.12
LRC	48.83±1.74	53.61±1.80	56.66±1.22	59.29±2.43
SRC	52.09±1.04	55.18±1.29	56.89±1.58	59.13±2.36
ProCRC	53.83±1.34	54.10±1.39	55.74±2.08	56.01±1.27
CBDS	56.78±2.11	60.92±1.72	62.30±1.65	64.56±1.63
DLSI	53.19±1.50	56.36±1.35	57.98±1.65	61.96±2.30
DSRM	53.42±1.39	58.52±1.68	62.70±1.42	66.39±1.21
DLSR	56.49±1.75	61.43±1.36	64.47±2.58	65.85±2.69
RLSL	58.11±1.09	62.26±1.58	66.50±1.39	68.66±1.79
RPCA	49.41±1.81	59.48±2.11	62.30±1.56	64.24±2.02
LatLRR	51.89±1.71	56.79±1.65	60.27±1.61	65.50±2.95
SLRM	54.77±1.87	59.57±0.85	61.56±1.73	63.44±2.77
DKSVD	51.64±1.04	54.03±1.68	56.31±1.91	59.55±1.36
LC_KSVD	52.43±1.78	55.74±1.05	58.47±1.57	60.84±1.25
LDA	40.72±1.21	43.61±1.86	46.11±1.85	51.44±1.29
PCE	51.09±1.83	53.93±1.75	56.35±1.04	59.56±2.05
SRRS	52.24±2.05	56.80±1.97	60.23±2.44	62.84±1.21
MSRL-G	56.69±2.22	62.00±1.56	64.52±1.58	66.94±2.21
MSRL	61.48±1.86	67.02±1.79	70.12±1.38	73.11±1.95
SMSRL	**62.69±1.42**	**67.93±1.56**	**70.86±1.30**	**74.45±2.13**

TABLE II: Recognition accuracies (mean±std %) of different methods on the COIL-100 database.

Alg.	10	15	20	25
LRLR	66.12±0.73	70.59±0.64	72.79±0.82	74.47±0.70
LRRR	65.98±0.94	70.11±0.65	73.22±0.71	75.64±0.59
SLRR	68.17±0.76	71.85±0.59	73.81±0.70	73.69±0.53
SVM	79.25±0.52	84.80±0.62	88.15±0.47	90.79±0.65
CapSVM	83.29±0.50	88.27±0.63	91.49±0.36	93.44±0.43
CRC	76.20±0.61	81.36±0.42	84.33±0.59	86.33±0.52
LLC	81.63±0.82	86.93±0.49	90.25±0.46	92.50±0.50
LRC	84.23±0.60	89.32±0.50	91.88±0.55	93.71±0.41
SRC	78.33±0.61	85.10±0.62	87.43±0.50	90.89±0.65
ProCRC	74.35±1.00	82.82±0.55	87.48±0.26	90.85±0.29
CBDS	71.46±0.54	78.56±0.37	79.28±0.36	82.65±0.89
DLSI	79.79±0.57	87.87±0.39	91.56±0.47	93.74±0.51
DSRM	82.97±0.48	88.23±0.52	91.10±0.42	92.93±0.37
DLSR	84.59±0.55	88.07±0.50	90.19±0.39	92.09±0.46
RPCA	82.56±0.65	88.31±0.87	91.72±0.31	93.53±0.35
RLSL	84.89±0.62	88.95±0.30	91.36±0.47	93.01±0.44
LatLRR	83.27±0.74	88.30±0.37	91.18±0.32	93.24±0.38
SLRM	83.72±0.75	88.86±0.22	91.56±0.36	93.80±0.45
DKSVD	78.99±0.67	83.80±0.54	86.51±0.59	88.53±0.31
LC_KSVD	79.99±0.65	85.15±0.56	87.94±0.60	90.13±0.36
LDA	53.55±0.44	69.56±0.35	79.13±0.91	84.53±0.60
PCE	78.61±0.31	84.86±0.65	88.08±0.42	90.81±0.77
SRRS	84.42±0.66	89.81±0.59	91.71±0.68	93.48±0.73
MSRL-G	86.48±0.48	91.18±0.34	94.06±0.37	95.74±0.44
MSRL	88.40±0.59	93.32±0.57	95.87±0.41	97.15±0.30
SMSRL	**93.71±0.40**	**96.50±0.31**	**97.70±0.49**	**98.38±0.31**

as the unlabeled samples of SMSRL. That is, the labels of all the training samples and the training and test features are simultaneously used as the inputs to train SMSRL. Specifically, the semantic labels of all the training samples are simultaneously employed to learn discriminative marginalized regression targets, which is the same as MSRL. Meanwhile, the adaptive probabilistic graph structure is trained to improve the compactness of the learned training and test representations on the projected semantic space.

We implement recognition experiments on diverse publicly available image datasets of four different types: 1) Object datasets including the Pittsburgh food image dataset (PFID) [40] and COIL-100 [41]; 2) Face image datasets consisting of Extended YaleB [42], CMU PIE [43] and AR [44]; 3) Texture image datasets including KTH-TIPS [45] and CurRet [46]; 4) The Fifteen-scene categories recognition dataset [47] for scene recognition. For each dataset, we randomly select several images from each class as training samples, and the remaining images are used for testing. All the experiments[1] are conducted ten times, and the average accuracies and standard deviations are reported.

B. Experiments for Object Recognition

To demonstrate the effectiveness of our method for handling the object recognition problem, we evaluate the performance of the proposed MSRL and SMSRL methods on the PFID and COIL-100 datasets. Descriptions of the two datasets are as follows:

The Pittsburgh Food Image Dataset (PFID): The PFID dataset is a released food recognition dataset, which is composed of fast food images and videos from chain restaurants. A subset of 61 categories of food items (e.g., McDonalds Big Mac) are used in our experiments. Each category of food is from three different restaurants, and each restaurant provides six images in six different viewpoints. That is, each category of food has eighteen images. We can see that this dataset is very difficult for recognition, and we employ the gray-scale PRICoLBP [48] for feature extraction. We randomly choose $6, 8, 10, 12$ images of each category as training samples, and the rest of images are used for testing.

The COIL-100 Dataset: The COIL-100 dataset includes different views of 100 objects under different lighting conditions. Each image is resized to 32×32 pixels and the challenge of this dataset is evaluated on alternative viewpoints. We randomly select 10, 15, 20, 25 images per object as training samples, and the remaining images are treated as test samples.

Each experiment is repeated 10 times, and the experimental results on the PFID and COIL-100 datasets of different methods are shown in Tables I and II, respectively. From both tables, it is easy to find that our methods can achieve the highest recognition results in comparison with all the compared algorithms. In most cases, the semi-supervised algorithm SMSRL can obtain higher results than MSRL. Compared with the related methods such as DSRM, SLRM and PCE, our method still has remarkable superiorities on this dataset. Our SMSRL also has obvious superiority in comparison with the semi-supervised SLRM. For Table II, when the number of the training samples is 20, at least 3.7% performance gain is obtained in comparison with the rest of algorithms, and our method achieves the encouraging average recognition accuracy as high as 97.15% when using 25 training samples.

[1]The MATLAB codes of our MSRL and SMSRL has been released at http://www.yongxu.org/lunwen.html.

TABLE III: Recognition accuracies (mean±std %) of different methods on the Extended YaleB database.

Alg.	10	15	20	25
LRLR	82.72±0.89	86.21±0.74	85.37±0.94	87.67±0.86
LRRR	83.22±1.52	87.26±1.45	89.29±0.71	90.59±0.80
SLRR	83.77±1.55	88.37±1.46	90.34±0.55	91.33±0.67
SVM	80.88±1.82	89.35±1.24	92.74±0.87	95.07±0.57
CapSVM	87.98±1.33	93.54±0.99	94.98±0.93	96.93±0.79
CRC	86.22±1.34	92.43±0.77	94.99±0.50	96.73±0.49
LLC	79.82±0.46	88.63±0.31	91.52±0.48	94.20±0.58
LRC	82.67±1.52	89.50±0.82	91.86±0.77	93.53±0.74
SRC	85.23±1.12	93.45±0.68	95.35±0.62	96.18±0.50
ProCRC	86.18±0.82	92.52±0.68	95.36±0.67	97.57±0.51
CBDS	85.59±1.44	93.18±1.19	95.53±0.80	96.46±0.62
DLSI	87.01±0.63	92.71±0.58	94.26±0.33	96.16±0.55
DSRM	89.08±0.84	92.95±0.78	94.62±0.59	96.54±0.46
DLSR	86.44±0.97	93.60±0.73	94.78±0.71	95.84±0.42
RLSL	88.41±0.78	93.16±0.50	94.66±0.18	95.56±0.30
RPCA	86.21±0.26	90.52±0.44	93.52±0.61	95.41±0.36
LatLRR	83.65±2.03	90.40±1.19	93.59±1.00	95.85±0.58
SLRM	83.70±1.47	89.92±1.19	93.10±0.97	95.28±0.86
DKSVD	83.67±0.77	85.58±0.32	89.50±0.26	92.34±0.71
LC_KSVD	84.15±1.79	89.59±0.93	93.24±0.69	94.34±0.72
LDA	83.19±1.13	86.14±1.06	89.00±1.63	91.54±1.28
PCE	84.78±1.50	90.69±1.27	93.49±0.68	95.57±0.85
SRRS	83.53±1.63	91.54±1.16	93.83±0.86	95.97±0.95
MSRL-G	86.92±1.16	92.96±0.92	95.50±0.91	97.23±0.68
MSRL	89.89±1.05	94.97±0.99	96.88±0.58	98.09±0.47
SMSRL	**93.58±1.21**	**97.29±0.78**	**98.41±0.29**	**99.09±0.22**

TABLE IV: Recognition accuracies (mean ± std %) of different methods on the CMU PIE database.

Alg.	10	15	20	25
LRLR	79.89±1.17	83.70±0.57	85.73±0.58	86.80±0.45
LRRR	82.55±0.84	86.98±0.83	89.19±0.65	90.23±0.84
SLRR	83.93±0.73	87.65±0.70	89.61±0.69	90.52±0.82
SVM	77.95±1.06	86.66±0.75	90.70±0.63	92.66±0.53
CapSVM	85.99±0.68	91.12±0.52	93.46±0.35	94.72±0.29
CRC	85.51±0.54	90.43±0.48	92.62±0.45	93.73±0.39
LLC	80.46±0.40	86.62±0.57	91.90±0.25	93.27±0.36
LRC	75.42±0.92	85.61±0.62	90.17±0.52	92.65±0.38
SRC	83.98±0.71	89.97±0.66	91.55±0.39	92.92±0.38
ProCRC	87.37±0.97	91.97±0.32	93.76±0.32	94.70±0.16
CBDS	81.74±0.92	88.33±0.82	91.37±0.55	93.21±0.66
DLSI	82.54±0.51	87.56±0.58	90.60±0.36	93.25±0.61
DSRM	85.60±0.61	90.94±0.46	93.08±0.35	94.49±0.52
DLSR	85.21±0.61	91.06±0.45	92.53±0.45	93.68±0.29
RLSL	87.25±0.64	91.43±0.41	93.22±0.37	94.38±0.33
RPCA	81.69±0.36	84.26±0.41	88.24±0.32	91.06±0.12
LatLRR	81.74±0.79	84.68±0.55	88.36±0.63	91.83±0.48
SLRM	84.24±0.73	88.60±0.62	91.74±0.63	93.24±0.53
DKSVD	81.83±0.86	88.86±0.73	91.77±0.34	93.69±0.29
LC_KSVD	83.62±0.67	89.66±0.68	92.44±0.34	93.95±0.31
LDA	77.78±0.53	81.86±0.70	83.86±0.88	90.07±0.73
PCE	83.87±0.54	87.76±0.50	88.69±0.52	88.98±0.33
SRRS	80.57±1.47	87.27±0.86	91.01±0.61	93.16±0.35
MSRL-G	85.47±0.85	91.19±0.68	93.18±0.44	94.61±0.31
MSRL	89.51±0.62	93.39±0.47	95.02±0.27	95.96±0.22
SMSRL	**90.23±0.63**	**93.78±0.37**	**95.49±0.28**	**96.25±0.23**

C. Experiments for Face Recognition

We apply MSRL and SMSRL to three real face recognition scenarios to evaluate the performance of our method.

The Extended YaleB Dataset: The extended YaleB dataset contains 2414 front face images from 38 individuals and each individual has around 64 images under various illumination conditions.The main challenge of this set is to deal with varying illumination conditions and expressions.

The CMU PIE Dataset: The CMU PIE face dataset includes 41,368 face images of 68 subjects. Our experiments are performed on the images under five poses (C05, C07, C09, C27 and C29), in which each subject has 170 images.

The AR Dataset: The AR face dataset contains about $4,000$ color face images of 126 subjects, which consist of the frontal faces with different illuminations, disguises and facial expressions. Each subject provides 26 images captured in two separate sessions under different conditions. In this experiment, we select a subset including 2600 images from 50 female and 50 male subjects. Similar to the implementation in [13], we project all the images onto 540-dimension with a randomly generated matrix from a zero-mean normal distribution.

For the Extended YaleB and CMU PIE datasets, we randomly select 10, 15, 20, 25 images for each subject as the training set, and regard the rest of the images as the test set. For the AR dataset, we randomly select $8, 11, 14, 17$ images of each subject as the training set, and the remaining images as the test set. The average recognition results on these datasets are respectively shown in Tables III, IV and V. It can be observed that the proposed MSRL and SMSRL methods achieve the highest recognition rates, which also verify that our methods are effective enough to yield promising recognition results. Specifically, SMSRL achieves about 3% improvement in comparison with other algorithms on different datasets.

TABLE V: Recognition accuracies (mean±std %) of different methods on the AR database.

Alg.	8	11	14	17
LRLR	76.75±1.37	88.93±0.86	93.02±0.63	94.92±0.68
LRRR	90.16±0.55	93.34±0.67	94.54±0.56	95.19±0.69
SLRR	88.61±0.57	92.23±0.72	93.89±0.46	95.45±0.84
SVM	80.74±1.58	85.59±1.15	92.00±0.78	95.21±0.95
CapSVM	89.25±0.75	94.37±0.76	96.37±0.63	97.71±0.64
CRC	86.48±0.92	91.67±0.62	94.29±0.53	95.59±0.62
LLC	81.01±1.17	86.03±1.29	89.71±1.03	92.18±0.95
LRC	77.17±1.53	85.62±0.97	90.68±1.07	93.98±1.04
SRC	83.74±0.99	89.59±1.10	93.14±0.61	95.19±0.80
ProCRC	89.31±0.62	93.52±0.50	95.48±0.49	96.61±0.60
CBDS	88.68±0.86	93.19±0.44	95.19±0.41	96.31±0.46
DLSI	78.78±1.02	85.93±1.01	89.92±0.76	93.17±0.97
DSRM	88.96±1.11	93.13±0.92	94.58±1.00	95.78±1.44
DLSR	87.76±1.42	93.68±0.88	94.36±0.62	95.18±0.46
RLSL	90.03±0.86	93.00±0.81	96.28±0.57	97.94±0.56
RPCA	77.32±1.43	84.39±1.33	88.82±0.90	92.62±0.77
LatLRR	87.85±1.36	93.71±0.87	95.49±0.47	96.13±0.50
SLRM	86.18±1.35	92.64±0.98	95.97±0.43	96.78±0.57
DKSVD	83.86±1.03	90.66±0.98	93.95±0.88	95.91±0.78
LC_KSVD	89.24±0.82	92.43±0.80	93.47±0.80	96.08±0.95
LDA	79.47±1.01	88.93±1.25	90.50±0.89	92.11±0.58
PCE	87.60±0.86	91.65±0.78	94.08±0.66	96.00±0.58
SRRS	84.20±1.14	90.17±1.25	94.11±1.24	96.17±0.68
MSRL-G	84.40±0.79	91.00±1.01	94.67±1.00	96.33±0.74
MSRL	91.97±0.81	95.33±0.64	96.83±0.46	97.89±0.65
SMSRL	**95.11±0.65**	**97.30±0.66**	**98.28±0.44**	**98.64±0.42**

D. Experiments for Texture Recognition

In this experiment, we evaluate the performance of the proposed algorithm on two widely used texture datasets, $i.e.$ the KTH-TIPS and CurRet datasets. The KTH-TIPS dataset includes 10 static texture categories, and each image is captured at nine scales, three different poses, and under three

TABLE VI: Recognition accuracies (mean±std %) of different methods on the KTH-TIPS database.

Alg.	Accuracy		Alg.	Accuracy
LLC	95.32±0.87		CapSVM	95.55±1.28
LLC*	95.71±1.44		ProCRC	93.83±1.45
LRC	90.41±2.26		RLSL	95.20±0.91
CRC	95.44±1.43		LDA	92.93±2.36
LRLR	82.90±1.45		CBDS	95.44±1.22
LRRR	82.78±1.28		DLSR	95.27±1.40
SLRR	82.12±2.22		SRC	93.77±0.99
RPCA	95.20±1.37		DKSVD	95.85±0.87
PCE	95.76±0.57		DSRM	95.63±1.23
LC_KSVD	96.01±1.15		SLRM	93.51±1.54
SRRS	95.56±1.11		MSRL-G	93.83±1.05
DLSI	96.00±2.00		MSRL	97.31±1.20
LatLRR	94.93±1.73		SMSRL	**97.88±0.91**

TABLE VII: Recognition accuracies (mean±std %) of different methods on the CUReT database.

Alg.	Accuracy		Alg.	Accuracy
LLC	95.63±0.87		CapSVM	96.94±0.33
LLC*	96.03±0.94		ProCRC	91.92±0.50
LRC	95.27±0.30		RLSL	96.90±0.34
CRC	92.28±0.43		LDA	91.93±0.45
LRLR	88.40±0.71		CBDS	88.36±0.39
LRRR	88.17±0.77		DLSR	95.80±0.43
SLRR	89.90±0.81		SRC	93.06±0.41
RPCA	93.67±0.61		DKSVD	92.59±0.58
PCE	90.64±0.44		DSRM	94.86±0.49
LC_KSVD	92.75±0.51		SLRM	95.89±0.30
SRRS	96.12±0.28		MSRL-G	98.34±0.21
DLSI	96.20±0.12		MSRL	98.72±0.18
LatLRR	95.86±0.29		SMSRL	**98.80±0.19**

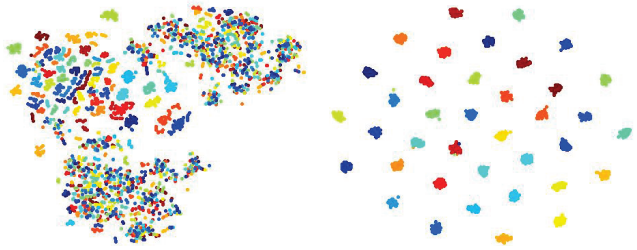
different illuminations. Each category has 81 samples.The CUReT data set is another popularly used texture recognition dataset. A subset containing 61 categories with 92 samples in each category is employed in our experiments. All the images are collected under different viewpoint directions and various illumination conditions. To obtain compact features of these texture images on both datasets, we utilize the PRICoLBP features [48] to construct our data representation learning model. For both texture datasets, we randomly select 40 samples per category to form the training set, and the remaining images are regarded as the test samples. The experimental results using different methods are summarized in Table VI and Table VII, respectively. As can be seen in both tables, our MSRL and SMSRL methods achieve competitive performance on both KTH-TIPS and CurRet datasets. It is worth noting that the proposed MSRL and SMSRL methods learn marginal representations and outperform the state-of-the-art algorithms, which reveal their strong capabilities in texture recognition.

E. Experiments for Scene Recognition

Scene recognition is a classic problem in computer vision. To evaluate the performance of the proposed MSRL and SMSRL methods in scene recognition, we implement our method on the fifteen scene categories dataset to demonstrate its effectiveness. This dataset contains totally 4485 indoor and outdoor scene images from 15 categories, such as livingroom and outdoor street. Instead of using the original features, we employ the features presented in [13] for recognition.

TABLE VIII: Average recognition accuracies (%) of different methods on the Fifteen scene categories database.

Alg.	Accuracy		Alg.	Accuracy
LLC	89.2		CapSVM	96.6
LRC	91.9		ProCRC	97.5
CRC	92.3		RLSL	98.1
LRLR	94.4		LDA	92.7
SRC	91.8		CBDS	95.7
LRRR	87.2		DLSR	95.9
SLRR	89.5		SVM	93.6
RPCA	92.1		DSRM	97.6
PCE	96.4		Lazebnik [47]	81.4
LC_KSVD	92.9		DKSVD	89.1
SRRS	97.9		MSRL-G	97.9
DLSI	92.4		MSRL	98.5
LatLRR	91.5		SMSRL	**99.0**



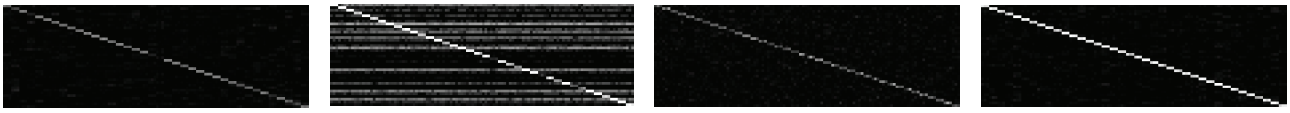
(a) Original features (b) Learned representation

Fig. 2: t-SNE visualization of (a) the original features and (b) the learned representation obtained by using MSRL on Extended YaleB, respectively. It is obvious that semantically similar categories of our method are distributed closely and otherwise faraway. From (b), we can clearly see that there are 38 categories.

Following the experimental protocols used in [13], [19], we randomly select 100 images of each category as training samples, and treat the rest of images as testing samples. To make fair comparisons, some experimental results are directly cited from literature [13]. Table VIII shows the average experimental accuracies by different methods. It is doubtless that our methods maintain the best performance and outperform all other algorithms. It should be noted that SMSRL achieves an overwhelmingly high average accuracy of 99.0%. Moreover, we notice that the proposed MSRL method can achieve very high recognition results for each category, and the worst recognition result is still as high as 96%, which indicates that our method is suitable to handle the scene recognition task.

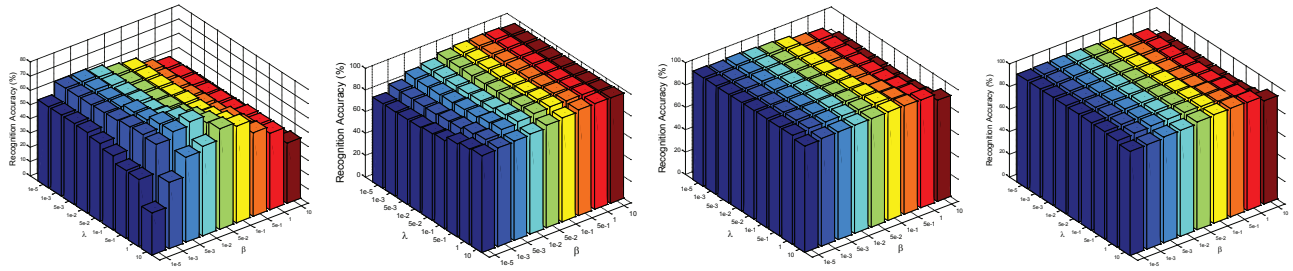
F. Experimental Analysis

From the above eight tables, we can see that the proposed MSRL and SMSRL achieve superior recognition performance on four different applications. Based on these experimental results, a number of interesting points are achieved as follows. (1) In most cases, the results demonstrate that, compared to the state-of-the-art data representation learning algorithms, our methods are consistently better on all the eight datasets. This testifies our claim that MSRL is capable of learning discriminative visual representations and improving the recognition accuracies. To clearly illustrate the discrimination of the learned representations, we randomly choose three images per



(a) SLRR (b) DLSR (c) SLRM (d) MSRL

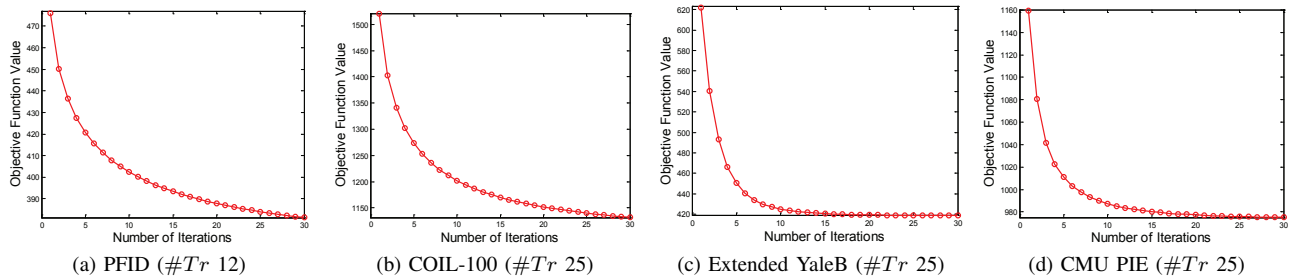
Fig. 1: 2D visualization of learned data representations of SLRR, DLSR, SLRM and MSRL.



(a) PFID ($\#Tr$ 12) (b) COIL-100 ($\#Tr$ 25) (c) Extended YaleB ($\#Tr$ 25) (d) CMU PIE ($\#Tr$ 25)

(e) AR ($\#Tr$ 17) (f) KTH_TIPS ($\#Tr$ 40) (g) CurRet ($\#Tr$ 40) (h) Fifteen Scene ($\#Tr$ 100)

Fig. 3: The performance evaluation (%) of MSLR versus parameters λ and β on eight datasets.



(a) PFID ($\#Tr$ 12) (b) COIL-100 ($\#Tr$ 25) (c) Extended YaleB ($\#Tr$ 25) (d) CMU PIE ($\#Tr$ 25)

(e) AR ($\#Tr$ 17) (f) KTH_TIPS ($\#Tr$ 40) (g) CurRet ($\#Tr$ 40) (h) Fifteen Scene ($\#Tr$ 100)

Fig. 4: Convergence curves of the proposed method on eight different databases.

subject from the Extended YaleB dataset, and 2D visualization of learned data representations are shown in Fig. 1. Obviously, the margins between different classes using our MSRL are highlighted in comparison with the related algorithms, $i.e.$ SLRR, DLSR and SLRM. Therefore, it is reasonable that the proposed MSRL gains better performance. (2) Compared with the linear regression methods, $i.e.$ LRLR, LRRR, SLRR and SLRM, MSRL and SMSRL make significant improvements. This manifests the necessity and advantages of learning predicted targets from the robust latent subspace and the adaptive probabilistic graph structure. The results also provide quantitative supports to our argument that MSRL encourages the optimal hidden information to be saturated in the learned data representations. (3) SLRM and SMSRL are better than

sparse and low-rank representation based methods as well as the dictionary learning methods. The main reason is that they focus on exploring the best reconstruction of the original data, which does not mean the best discrimination. This result clearly demonstrates that the discriminative information coded by the marginal constraint, and the graph regularizer contributes positively in improving intra-class compactness. Fig. 2 shows the t-SNE visualization of the original and learned features, respectively. (4) DLSR, MSRL and SMSRL achieve promising results due to the relaxation of the regression targets. However, the marginal targets of MSRL and SMSRL are directly learned from the data other than the binary regression targets, and the graph structure adaptation provides a preferable solution of addressing the over-fitting issue. This proves that both properties are important for learning marginal visual representations. (5) SLRM, DSRM, MSRL and SMSRL are superior to the other methods in general, which indicates the effectiveness of the graph structure learning on the predicted targets such that visually similar data can implicitly share common targets and are scattered together. (6) In general, the semi-supervised methods have clear tendency to achieve better recognition results. This demonstrates that interpolation of the unlabeled information on dynamical graph structure design can enhance the subtle distinction of the learned representations on the projected semantic space. In the training stage, MSRL only enforces that when two instances from the training samples are close in the same semantic space, they should have a much higher possibility to be connected in the adaptive probabilistic graph learning. However, the test samples are not taken into account of the learning process, but directly take $\boldsymbol{W}\hat{\boldsymbol{X}}$ as the test representation. In contrast, based on the experimental settings, SMSRL not only ensures the optimal mapping matrix from the training samples to discriminative target space, but also confirms the probabilistic connectivity between the training and test samples in the learned new representations using the adaptive probabilistic connectivity on the whole dataset. Among the global graph embedding learning processing, the connective relationship between the learned training representation $\boldsymbol{W}\boldsymbol{X}$ and testing representation $\boldsymbol{W}\hat{\boldsymbol{X}}$ is fully considered, which is the main reason why SMSRL is superior to MSRL.

Overall, SLRM and SMSRL learn marginal visual representations by seamlessly incorporating the robust latent subspace learning, probabilistic graph structure adaptation and flexibly marginalized regression construction into a unified framework. It is observed that SLRM and SMSRL can achieve the state-of-the-art performance, which shows that the learned representations of our methods are distinctive and discriminative for recognition.

G. Parameter Sensitivity

In this section, we examine the parameter sensitivity of MSRL, and there are several regularization parameters to be tuned in our proposed framework. In our experiments, the dimensionality s of the latent subspace should be lower than the full-rank of the data to preserve the low-rank fitness, and is tuned in the range of $[c/2, c)$ where c is the number of classes. The experimental results show that the value of s does not greatly influence the outputs, and we empirically set the value of s near to c-1. For simplicity, we directly set the parameter γ=0.05 in our experiments. Here we concentrate on discussing the influence of parameters λ and β by examining the variability of MSRL recognition performance with different values of both parameters. These two parameters are tuned from the candidate set $\{0.00001, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0, 10.0\}$, and the recognition results versus the values of parameters on eight datasets are shown in Fig. 2, where $Tr(\#)$ means the number of training samples per class. It can be easily observed that the performance variations are different on respective datasets, but our MSRL method is not sensitive to the values of the regularization parameters when they are not very large. This also indicates that the probability graphical structure adaptation and the elastic-net regularization are both critical and indispensable to marginal data representation learning.

H. Convergence Study

Based on Proposition 1 and Theorem 1, the proposed model in (16) is convex with respect to each variable, and the convergence property of the proposed optimization algorithm is theoretically guaranteed. In this section, we experimentally verify convergence nature of the proposed optimization algorithm on eight datasets. The convergence curves are shown in Fig. 3. We notice that Algorithm 2 performs well in terms of convergence, and the objective function of MSRL monotonically decreases with respect to the number of iterations. It is apparent that for all the eight datasets, the objective function of MSRL becomes relatively stable within 30 iterations, which also justifies the effectiveness of the proposed optimization algorithm.

I. Time Comparison

To explicitly show the computational complexity of the proposed method, we compare the efficiency of competing methods. The Matlab codes of all algorithms are obtained from the corresponding authors, and all algorithms were implemented in MATLAB on a 3.30-GHz CPU Windows 7 machine with 8 GB memory. As an example, we perform experiments on the Extended YaleB dataset, and randomly select 25 images per subject as training samples and the rest as testing samples. The run time comparisons of different algorithms with respect to the training and test time are listed in Table IX. It should be pointed out that LLC, LRC, CRC, SRC and DSRM have no training time. Table IX manifests that the proposed MSRL is very efficient as the seventh fastest algorithm among the 22 competing methods, while the performance of MSRL is greatly superior to the faster algorithms. Consequently, the proposed MSRL framework not only achieves highest recognition accuracies but also enjoys high efficiency in comparison with the competing methods.

VII. Conclusion

In this paper, an effective marginal visual representation learning framework was proposed based on marginal regression targets learning, robust latent subspace construction

TABLE IX: Run time comparisons of different algorithms (s).

Alg.	Train	Test		Alg.	Train	Test
LLC	—	42.83		PCE	1.39	0.30
LRC	—	59.50		DLSI	9.29	44.41
CRC	—	43.39		CBDS	153.90	1.71
SRC	—	899.55		DLSR	5.56	0.54
CapSVM	6.43	0.26		LDA	3.95	0.32
LRLR	3.74	0.13		DKSVD	76.72	0.53
LRRR	2.58	0.14		RLSL	65.44	4.23
SLRR	20.44	0.16		LC_KSVD	64.50	0.84
RPCA	89.43	0.61		SLRM	10.97	6.81
LatLRR	128.80	0.63		DSRM	—	92.67
SRRS	205.18	2.90		MSRL	7.07	0.20

and probabilistic graph structure adaptation. It seamlessly incorporates the local and global consistencies on regression targets into a common framework that tackles the problem of data representation. The marginal targets learning from data provide sufficient flexibilities of fitting regression tasks. Moreover, the underlying latent information of data is explored to make targets prediction. The learned data representations are more informative and discriminative in comparison with other representations mentioned in this paper. The resulting problem was efficiently solved by an iterative optimization strategy, in which its convergence property is demonstrated from both theoretical and experimental perspectives. In addition, the experimental results on eight datasets have demonstrated that our method outperforms the state-of-the-art data representation algorithms, which shows the efficacy of the proposed MSRL method. On the other hand, our method can be easily extended to the problem of semi-supervised data representation learning, where information of the unlabeled samples is embedded into the graph structure learning. Thus, the proposed MSRL framework of marginal visual representation learning can be used for both supervised and semi-supervised discriminative data representation learning.

REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798-1828, 2013.

[2] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognit. Neurosci.*, vol. 3, no. 1, pp. 71-86, 1991.

[3] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711-720, 1997.

[4] Z. Fan, Y. Xu, and D. Zhang D. "Local linear discriminant analysis framework using sample neighbors," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1119-1132, 2011.

[5] J. Wright, A.Y. Yang, A. Ganesh, et al, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210-227, 2009.

[6] Y. Xu, Z. Zhong, J. Yang, J. You, D. Zhang, "A New Discriminative Sparse Representation Method for Robust Face Recognition via L2 Regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2233-2242, 2017.

[7] Z. Zhang, Y. Xu, J. Yang, X. Li, D. Zhang, "A survey of sparse representation: algorithms and applications," *IEEE Access*, vol. 3, pp. 490-530, 2015.

[8] G. Liu, Z. Lin, S. Yan, et al. "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171-184, 2013.

[9] Z. Zhang, Y. Xu, L. Shao, J. Yang, "Discriminative Block-Diagonal Representation Learning for Image Recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, DOI: 10.1109/TNNLS.2017.2712801, 2017.

[10] S. Li, Y. Fu, "Learning Robust and Discriminative Subspace with Low-Rank Constraints," *IEEE Trans. Neural Netw. Learn. Syst.*. vol. 27, no. 11, pp. 2160-2173, 2016.

[11] M. Aharon, M. Elad, A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311-4322, 2006.

[12] Q. Zhang, B. Li, "Discriminative K-SVD for dictionary learning in face recognition," *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2691-2698, 2010.

[13] Z. Jiang, Z. Lin, L.S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651-2664, 2013.

[14] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Sci.*, vol. 290, no. 5500, pp. 2323-2326, 2000.

[15] X. He and P. Niyogi, "Locality preserving projections," in Proc. of *Neural Inf. Process. Syst*, pp. 153-160, 2003.

[16] X. Peng, J. Lu, Z. Yi, Y. Rui, "Automatic Subspace Learning via Principal Coefficients Embedding," in Proc. of *IEEE Trans. on Cybern.*, vol. 47, no. 11, pp. 3583-3596, 2017.

[17] J. Guo, Y. Guo, X. Kong, R. He, "Unsupervised Feature Selection with Ordinal Locality," in Proc. of *IEEE Int. Conf. Multimedia Expo*, pp. 1213-1218, 2017.

[18] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in Proc. of *IEEE Int. Conf. Comput. Vis.*, pp. 471-478, 2011.

[19] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in Proc. of *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3360-3367, 2010.

[20] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2106-2112, 2010.

[21] E. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?"" *J. ACM*, vol. 58, no. 3, pp. 11, 2011.

[22] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in Proc. of *IEEE Int. Conf. Comput. Vis.*, pp. 1615-1622, 2011.

[23] M. Yin, J. Gao, Z. Lin, "Laplacian regularized low-rank representation and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 504-517, 2016.

[24] X. Cai, C. Ding, F. Nie, and H. Huang, "On the equivalent of low-rank linear regressions and linear discriminant analysis based regressions," in Proc. of *ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 1124-1132, 2013.

[25] Z. Li, Z. Lai, Y. Xu, J. Yang, D. Zhang, "A Locality-Constrained and Label Embedding Dictionary Learning Algorithm for Image Classification," *IEEE Trans. Neural Netw. Learn. Syst.*,vol. 28, no. 2, pp. 278-293, 2017, Doi: 10.1109/TNNLS.2015.2508025

[26] C. Wei, C. Chen, and Y. Wang, "Robust Face Recognition With Structurally Incoherent Low-Rank Matrix Decomposition," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3294-3307, 2014.

[27] Y. Li, J. Liu, Z. Li, Y. Zhang, H. Lu, and S. ma, "Learning low-rank representations with classwise block-diagonal structure for robust face recognition," *AAAI Conf. Artif. Intell.*, pp. 2810-2816, 2014.

[28] A. Golts, M. Elad, "Linearized kernel dictionary learning," *IEEE J. Sel. Top. Signal Process.*, vol. 10, no. 4, pp. 726-739, 2016.

[29] S. Xiang, F. Nie, G. Meng, C. Pan, C. Zhang, "Discriminative least squares regression for multiclass classification and feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 11, pp. 1738-1754, 2012.

[30] Z. Zhang, Z. Lai, Y. Xu, L. Shao, J. Wu, G. Xie, "Discriminative Elastic-Net Regularized Linear Regression," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1466-1481, 2017.

[31] X.-Y. Zhang, L. Wang, S. Xiang, C.-L. Liu, "Retargeted Least Squares Regression Algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2206-2213, 2015.

[32] C. Chang, C. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Sys. Technol.*, vol. 2, no. 3, pp. 27, 2011.

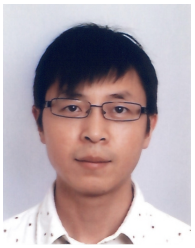
[33] F. Nie, X. Wang, H. Huang, "Multiclass Capped Lp-Norm SVM for Robust Classifications," in Proc. of *AAAI Conf. Artif. Intell.*, pp. 2415-2421, 2017.

[34] L. Jing, L. Yang, J. Yu, M. Ng, "Semi-supervised low-rank mapping learning for multi-label classification," in Proc. of *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1483-1491, 2015.

[35] F. Nie, X. Wang, H. Huang, "Clustering and projected clustering with adaptive neighbors," in Proc. of *ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 977-986, 2014.

[36] F. Nie, W. Zhu, X. Li, "Unsupervised Feature Selection with Structured Graph Optimization," in Proc. *AAAI Conf. Artif. Intell.*, pp. 1302-1308, 2016.
[37] O. Koyejo, S. Acharyya, J. Ghosh, "Retargeted matrix factorization for collaborative filtering," in Proc. *ACM Conf. Recomm. Syst.*, pp. 49-56, 2013.
[38] S. Cai, L. Zhang, W. Zuo, et al. "A probabilistic collaborative representation based approach for pattern classification," in Proc. of *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2950-2959, 2016.
[39] X. Fang, S. Teng, Z. Lai, et al. "Robust Latent Subspace Learning for Image Classification," *IEEE Trans. Neural Netw. Learn. Syst.*, 2017, DOI: 10.1109/TNNLS.2017.2693221.
[40] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang, "PFID: Pittsburgh fast-food image dataset," in Proc. *IEEE Int. Conf. Image Process.*, pp. 289-292, 2009.
[41] S. A. Nene, S. K. Nayar and H. Murase, "Columbia Object Image Library (COIL-100)," *Technical Report*, CUCS-006-96, 1996.
[42] A. Georghiades, P. Belhumeur, D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643-660, 2001.
[43] T. Sim, S. Baker, M. Bsat, "The CMU pose, illumination, and expression (PIE) database," in Proc. of *IEEE Int. Conf. Autom. Face Gesture Recognit.*, pp. 46-51, 2002.
[44] A. Martinez and R. Benavente, "The AR face database," *CVC Tech. Report* No. 24, 1998.
[45] E. Hayman, B. Caputo, M. Fritz, and J. Eklundh, "On the significance of real-world conditions for material classification," in Proc. *Eur. Conf. Comput. Vis.*, pp. 253-266, 2004.
[46] K. Dana, B. Van Ginneken, S. Nayar, and J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Trans. Graph.*, vol. 18, pp. 1-34, 1999.
[47] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in Proc. of *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2169-2178, 2006.
[48] X. Qi, R. Xiao, C. Li, J. Guo, X. Tang, "Pairwise rotation invariant co-occurrence local binary pattern," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2199-2213, 2014.

Zheng Zhang received the B.S degree from Henan University of Science and Technology and M.S degree from Shenzhen Graduate School, Harbin Institute of Technology (HIT) in 2012 and 2014, respectively. Currently, he is pursuing the Ph.D. degree in computer science and technology at Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China.



Ling Shao is currently a Full Professor and the Head of the Computer Vision and Artificial Intelligence Group with the Department of Computer Science and Digital Technologies, Northumbria University, Newcastle upon Tyne, and an Advanced Visiting Fellow with the Department of Electronic and Electrical Engineering, University of Sheffield. His research interests include computer vision, image processing, pattern recognition, and machine learning. He is a fellow of the British Computer Society and IET, and a Life Member of ACM. He is an Associate Editor of the IEEE Transactions on Image Processing, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Cybernetics and several other journals.

Yong Xu was born in Sichuan, China, in 1972. He received his B.S. degree and M.S. degree at Air Force Institute of Meteorology (China) in 1994 and 1997, respectively. He received the Ph.D. degree in Pattern recognition and Intelligence System at the Nanjing University of Science and Technology (NUST) in 2005. Now, he works at Shenzhen Graduate School, Harbin Institute of Technology. His current interests include pattern recognition, biometrics, machine learning and video analysis.

Li Liu received the B.Eng. degree in electronic information engineering from Xian Jiaotong University, Xi'an, China, in 2011, and the Ph.D. degree from the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, U.K., in 2014. He is currently a Research Fellow with University of East Anglia, U.K. His current research interests include computer vision, machine learning, and data mining.

Jian Yang received the B.S. degree in mathematics from the Xuzhou Normal University in 1995. He received the M.S. degree in applied mathematics from the Changsha Railway University in 1998 and the Ph.D. degree from the Nanjing University of Science and Technology (NUST), on the subject of pattern recognition and intelligence systems in 2002. In 2003, he was a postdoctoral researcher at the University of Zaragoza. From 2004 to 2006, he was a Postdoctoral Fellow at Biometrics Centre of Hong Kong Polytechnic University. From 2006 to 2007, he was a Postdoctoral Fellow at Department of Computer Science of New Jersey Institute of Technology. Now, he is a professor in the School of Computer Science and Technology of NUST. He is the author of more than 80 scientific papers in pattern recognition and computer vision. His journal papers have been cited more than 1600 times in the ISI Web of Science, and 2800 times in the Web of Scholar Google. His research interests include pattern recognition, computer vision and machine learning. Currently, he is an associate editor of Pattern Recognition Letters and IEEE TRANSACTION ON NEURAL NETWORKS AND LEARNING SYSTEMS, respectively.