Multiple Marginal Fisher Analysis

Zhenyu Huang, Hongyuan Zhu, Joey Tianyi Zhou, and Xi Peng, *Member, IEEE*

***Abstract*—Dimension reduction is a fundamental task of machine learning and computer vision, which is widely used in a variety of industrial applications. Over past decades, a lot of unsupervised and supervised algorithms have been proposed. However, few of them can automatically determine the feature dimension that could be adaptive to different data distributions. To obtain a good performance, it is popular to seek the optimal dimension by exhaustively enumerating some possible values. Clearly, such a scheme is ad hoc and computationally extensive. Therefore, a method which can automatically estimate the feature dimension in an efficient and principled manner is of significant practical and theoretical value. In this paper, we propose a novel supervised subspace learning method called multiple marginal Fisher analysis (MMFA), which can automatically estimate the feature dimension. By maxing the interclass separability among marginal points while minimizing within-class scatter, MMFA obtains low-dimensional representations with outstanding discriminative properties. Extensive experiments show that MMFA not only outperforms other algorithms on clean data, but also show robustness on corrupted and disguised data.**

***Index Terms*—Automatic dimension reduction, graph embedding, manifold learning, supervised subspace learning.**

I. Introduction

IN THE era of big data, it is challenging and crucial to develop effective and efficient methods to explore the latent value from massive data. However, this is a daunting task due to the increasing dimension of data accompanying very sparse useful information along with a large number of unwanted redundancy and noises [1]. Besides, the high dimension also brings an extra computational overhead, i.e., the so-called *curse of dimension*.

In the past decades, many dimension reduction algorithms [2]–[6] have been proposed to solve the curse of dimension. In general, existing algorithms can be roughly classified into unsupervised methods and supervised methods. Unsupervised methods aim to find a low-dimensional representation of original data without utilizing any label information. The most popular method is probably the principal components analysis (PCA) [2], which preserves the global structure of data with the maximum variance. More recently, the manifold learning methods are proposed to achieve the nonlinear dimension reduction, and typical works include ISOMAP [7], locally linear embedding [3], neighborhood preserving embedding (NPE) [8], Laplacian eigenmaps [9], locality preserving projections (LPP) [10], and their variants [11]–[16]. The key idea of them is to utilize the local manifold structure embedded in the high-dimensional space. The other well-known methods include sparsity preserving projections [17] and L1-graph [18]. Recently, Peng *et al.* [19], [20] theoretically discovered the connections between nuclear norm and Frobenius norm. Based on the Frobenius-norm representation, the principal coefficients embedding (PCE) method [21] was proposed and has achieved state-of-the-art performance in unsupervised subspace learning. Supervised methods utilize the label information to obtain more discriminative features. The most representative method is perhaps the linear discriminative analysis (LDA) [22], [23], and its variant [24], which aims to minimize the within-class scatter while maximizing the between-class scatter. In [5], Yan *et al.* showed that most of the aforementioned methods [25]–[28] can be unified into the graph embedding framework. Based on this framework, they proposed a new algorithm called marginal Fisher analysis (MFA) [5], which maximizes the separability between pairwise marginal data points.

Despite the success of these methods, most of them cannot estimate the dimension of feature space in a data-driven way. In general, they obtain the optimal feature dimension by exhaustively enumerating all possible values based on the classification accuracy. Clearly, such a strategy is computationally extensive and may cause the overfitting problem. Recently, some methods have been proposed to solve this problem, e.g., PCE [21] and MFA [5]. PCE reduces the dimension without the help of label information, which could also automatically estimate the dimension. Though PCE achieves impressive results, it is highly desirable to develop supervised automatic dimension reduction method. In practice, however, only a few efforts have been devoted. Under the framework of graph embedding [5], MFA is proposed, which builds two graphs

Manuscript received June 1, 2018; revised August 14, 2018; accepted September 2, 2018. Date of publication September 28, 2018; date of current version July 31, 2019. The work of Z. Huang and X. Peng was supported in part by the Fundamental Research Funds for the Central Universities under Grant YJ201748 and in part by the NFSC under Grant 61806135, Grant 61432012, and Grant U1435213. The work of J. T. Zhou was supported by RIE2020 Plan under Grant A1687b0033. *(Corresponding author: Xi Peng.)*

Z. Huang and X. Peng are with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: zyhuang.gm@gmail.com; pengx.gm@gmail.com).

H. Zhu is with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore 138632 (e-mail: zhuh@i2r.a-star.edu.sg).

J. T. Zhou is with the Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore 138632 (e-mail: joey.tianyi.zhou@gmail.com).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org. This includes a PDF that shows the detailed proof for Lemma 1 and Theorem 1, as well as the used notation and some sample images used in the experiments. This material is 380 KB in size.

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

Digital Object Identifier 10.1109/TIE.2018.2870413

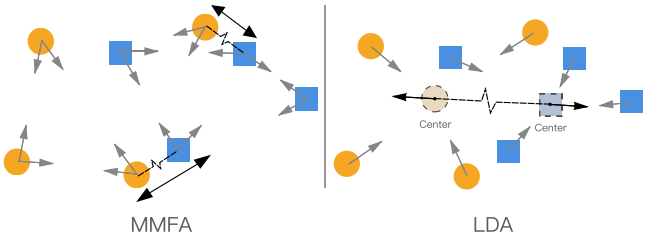

Fig. 1. Toy example to show the difference between LDA and our MMFA. In LDA, the data points move toward to the data center. And, the data center of each class keep away from the center of all data. In this binary class example, they keep away from each other. Thus, only the data follows Gaussian distribution, LDA succeeds in separating the different classes. MMFA solves this problem by considering only the marginal data points. As we can see, the data points move toward to their neighbors in the same class and only the marginal points keep away from their neighbors in different classes.

based on the marginal data points with the help of labels. One major advantage of MFA is that the feature dimension could be determined by using the number of between-class marginal pairs. However, MFA does not give mathematical detail on the feature dimension range; hence, it is more like a heuristic method. In fact, MFA and its variants [29] barely explore the connection between feature dimension and the the number of between-class marginal pairs in theory.

In this paper, we propose a novel supervised dimension reduction method called *Multiple Marginal Fisher Analysis* (MMFA), which could enjoy the advantage of automatic dimension estimation. Unlike the well-known LDA which assumes that data points follow the multivariate Gaussian distribution [30], [31], MMFA estimates the feature dimension using the marginal data points and the local consistence (i.e., manifold structure), thus, avoiding the requirement of data distribution assumption and enjoying promising performance in practical applications. A comparison between LDA and MMFA is shown in Fig. 1. Furthermore, different from other Fisher rule based methods such as MFA, our method could solve the *class-isolation issue*, i.e., when some classes are isolated from the others, the embeddings may overlap in the feature space and ignore the separability between the isolated classes and other classes. More details could refer to Fig. 2. In order to overcome this disadvantage, MMFA constructs the between-class graph by considering the multiple marginal data pairs which are also shown in Fig. 2.

Notations: For ease of presentation, we first define the used mathematical notation through this paper. To be exact, the lower-case letters denote scalars, the lower-case bold letters denote the vectors, and the upper-case bold ones denote matrices. Besides, for a given matrix $\mathbf{A}$, $r(\mathbf{A})$ denotes the rank of $\mathbf{A}$ and $Tr(\mathbf{A})$ denotes the trace of $\mathbf{A}$.

Organization: The rest of this paper is organized as follows. In Section II, we briefly introduce some related works. Section III introduces our proposed MMFA. The experimental results are shown in Section IV. Finally, we give the conclusion and further discussion of this paper in Section V.

II. Related Works

In this section, we briefly introduce some related works for dimension reduction including unsupervised method PCE [21] and supervised methods including LDA [23] and MFA [5].

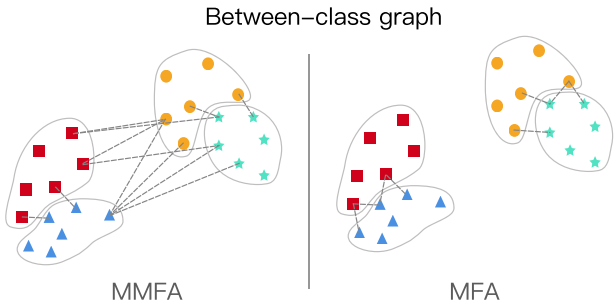

Fig. 2. Toy example to show the difference between MFA and our MMFA. One major advantage of MMFA over MFA is that the so-called *class-isolation issue* is addressed by the former. More specifically, MFA only connects the neighbors in different classes of marginal data points. As a result, the connections may only exit into the closest classes. As shown in the figure, in the between-class graph of MFA, there are no edges between red (blue) points and yellow (green) points which come from two far away classes. As a result, MFA will only try to separate the red (yellow) and blue (green) classes, while ignoring the separability of red-blue and yellow-green class. This probably lead to the mixture of between-class data points in the feature space and suboptimal results. Different from MFA, our MMFA addresses this issue by considering the connections between all pairwise classes, i.e., multiple marginal points.

A. Principal Coefficients Embedding (PCE)

Recently, Peng *et al.* [19] have shown that Frobenius-norm-based representation could enjoy the low-rank structure owned by nuclear-norm-based representation. Based on this theoretical study, Peng *et al.* [21] proposed a novel unsupervised subspace learning method called principal coefficient embedding, which could achieve both robustness and automatic dimension estimation.

For a given data $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\}$, PCE aims to removes the noise $\mathbf{E}$ from $\mathbf{X}$ to recover the clean data with self-representation regularization of $\mathbf{X}_0$. The objective function is given as follows:

$$\begin{aligned} \min_{\mathbf{C},\mathbf{X}_0,\mathbf{E}} \quad & \frac{1}{2}\|\mathbf{C}\|_F^2 + \frac{\lambda}{2}\|\mathbf{E}\|_F^2 \\ \text{s.t.} \quad & \underbrace{\mathbf{X} = \mathbf{X}_0 + \mathbf{E}}_{\text{Robustness}}, \underbrace{\mathbf{X}_0 = \mathbf{X}_0\mathbf{C}}_{\text{Self-expression}} \end{aligned} \tag{1}$$

where $\mathbf{C}$ denotes the representation matrix, which is obtained by performing SVD on the original data.

After obtaining the representation $\mathbf{C}$, PCE yields the low-dimensional features by embedding $\mathbf{C}$ into the feature space as an invariance. Although PCE has achieved state-of-the-art performance in image feature extraction, it does not utilize available label information to boost the performance for classification tasks.

B. Linear Discriminant Analysis (LDA)

Different from PCE, LDA [23] is a supervised subspace learning method, which aims to learn a space in which within-class data points (i.e., the points belonging to the same class) are as close as possible and between-class data points (i.e., the points belonging to different classes) are as far as possible.

For a given data $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\}$ distributed over the classes $\{c_0, c_1, \ldots, c_{n_c}\}$, LDA obtains the low-dimensional features $\mathbf{Y}$ with the help of the learned projection matrix $\mathbf{A}$

via $\mathbf{Y} = \mathbf{A}^T\mathbf{X}$. The objective function is as follows:

$$\arg\max_{\mathbf{A}} \quad \frac{\mathbf{A}^T\mathbf{S}_B\mathbf{A}}{\mathbf{A}^T\mathbf{S}_W\mathbf{A}} \tag{2}$$

where $\mathbf{S}_B$ and $\mathbf{S}_W$ denote the between-class and within-class scatter matrix, respectively, with the following definition:

$$\begin{aligned}\mathbf{S}_B &= \sum_{i=1}^{n_c} N_i(\hat{\mathbf{x}}_i - \hat{\mathbf{x}})(\hat{\mathbf{x}}_i - \hat{\mathbf{x}})^T \\ \mathbf{S}_W &= \sum_{i=1}^{n_c}\sum_{\mathbf{x}_k\in\mathbf{X}_i}(\mathbf{x}_k - \hat{\mathbf{x}}_i)(\mathbf{x}_k - \hat{\mathbf{x}}_i)^T\end{aligned} \tag{3}$$

where $\hat{\mathbf{x}}$ denotes the mean vector of $\mathbf{X}$, $\mathbf{X}_i$ denotes the data set belonging to the class c_i whose mean vector is $\hat{\mathbf{x}}_i$, and N_i is the number of samples in $\mathbf{X}_i$.

LDA leans discriminative features by utilizing within-class similarity $\mathbf{S}_W$ and between-class separability $\mathbf{S}_B$. In theory, the maximal feature dimension of LDA is $n_c - 1$ due to the rank of matrix $\mathbf{S}_B$ is less than $n_c - 1$. Thus, it would lead inferior performance for a large-scale dataset since $n_c - 1$ features may be insufficient to keep crucial information of the input space as explained in [23].

C. Marginal Fisher Analysis

Yan *et al.* [5] have shown that most dimension methods can be unified into a graph embedding framework. Under this framework, the dimension reduction methods obtain low-dimensional features by preserving the graph geometric structure from input space into a feature space. Along with this framework, a new supervised algorithm called MFA was proposed, of which major novelty lies on constructing a between-class and within-class graph as follows.

1) Within-class: $W_{ij} = W_{ji} = 1$ if $\mathbf{x}_j$ is among the k_1 nearest neighbors (NNs) of $\mathbf{x}_i$ in the same class.
2) Between-class: $W'_{ij} = W'_{ji} = 1$ if $(\mathbf{x}_i, \mathbf{x}_j)$ is among the k_2 shortest pairs among the set $\{(\mathbf{x}_i, \mathbf{x}_j)|\mathbf{x}_i \in \mathbf{X}_c, \mathbf{x}_j \notin \mathbf{X}_c\}$.

where $\mathbf{W}$ and $\mathbf{W}'$ are the affinity matrixes which denote the similarity of within-class and separability of between-class, respectively. MFA minimizes the similarity and simultaneously maximizes the separability in the low-dimensional space like LDA.

III. Multiple Marginal Fisher Analysis

In this section, we propose the MMFA, which enjoys three advantages, namely, data-adaptive feature dimension estimation, discriminative feature thanks to available data annotation, and a provable feature dimension lower bound.

A. Multiple Marginal Fisher Analysis

Most of the dimension reduction methods could be regarded as preserving the geometric structure and label information which correspond to an affinity graph and penalty graph, respectively. As discussed in Fig. 1, LDA suffered from the limitation of the Gaussian distribution assumption. Then, MFA [5] was proposed to solve this limitation by characterizing the between-class separability which only depends on the marginal data points. However, MFA suffers from the class-isolation issue as shown in Fig. 2. Hence, we propose a novel dimension reduction method called MMFA, which not only applies to the nonGaussian cases but also solves the class-isolated issue.

For a given $\mathbf{x}_i$, we define the corresponding low-dimensional feature $\mathbf{y}_i$ with the projection matrix $\mathbf{A}$ as follows:

$$\mathbf{y}_i = \mathbf{A}^T\mathbf{x}_i. \tag{4}$$

A certain criterion motivated by LDA [23] is to minimize the within-class similarity and maximize the between-class separability in the low-dimensional space. In MMFA, we characterize the within-class similarity in the embedding space by following [9]:

$$\begin{aligned}S_W &= \sum_i^n\sum_j^n \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij} \\ &= \sum_i^n\sum_j^n \left(\mathbf{y}_i^T\mathbf{y}_i - 2\mathbf{y}_i^T\mathbf{y}_j + \mathbf{y}_j^T\mathbf{y}_j\right) W_{ij} \\ &= \sum_i^n \left(\sum_j^n W_{ij}\right)\mathbf{y}_i^T\mathbf{y}_i + \sum_j^n\left(\sum_i^n W_{ij}\right)\mathbf{y}_j^T\mathbf{y}_j \\ &\quad - 2\sum_i^n\sum_j^n \mathbf{y}_i^T\mathbf{y}_j W_{ij} \\ &= 2\sum_i^n D_i\mathbf{y}_i^T\mathbf{y}_i - 2\sum_i^n\sum_j^n \mathbf{y}_i^T\mathbf{y}_j W_{ij} \\ &= 2Tr(\mathbf{Y}^T\mathbf{D}\mathbf{Y}) - 2Tr(\mathbf{Y}^T\mathbf{W}\mathbf{Y}) \\ &= 2Tr(\mathbf{A}^T\mathbf{X}(\mathbf{D} - \mathbf{W})\mathbf{X}^T\mathbf{A}).\end{aligned} \tag{5}$$

Furthermore, the between-class separability S_B is characterized by the multiple marginal pairs as follows:

$$\begin{aligned}S_B &= \sum_i^n\sum_j^n \|\mathbf{y}_i - \mathbf{y}_j\|^2 W'_{ij} \\ &= 2Tr(\mathbf{A}^T\mathbf{X}(\mathbf{D}' - \mathbf{W}')\mathbf{X}^T\mathbf{A})\end{aligned} \tag{6}$$

where $\mathbf{D}$ and $\mathbf{D}'$ are defined as

$$D_{ii} = \sum_j W_{ij}, \quad D'_{ii} = \sum_j W'_{ij} \tag{7}$$

where the $\mathbf{W}$ and $\mathbf{W}'$ are computed from the within-class and between-class data points as follows.

1) *Within-class graph*: We put an edge on the data points $\mathbf{x}_i$ and $\mathbf{x}_j$ if $\mathbf{x}_j$ is among the k_1 NNs of $\mathbf{x}_i$

$$W_{ij} = \begin{cases} \|\mathbf{x}_i - \mathbf{x}_j\|^2, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are connected in} \\ & \text{the within-class graph.} \\ 0, & \text{otherwise.}\end{cases} \tag{8}$$

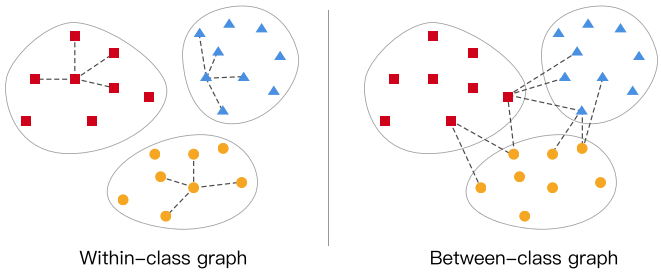

Fig. 3. Illustration on the graph construction of MMFA, where $k_1 = 4$ for the within-class graph and $k_2 = 3$ for the between-class graph. Note that, the within-class graph is built by the nearest neighbors and the between-class graph is built by the shortest pairs among every two classes.

2) *Between-class graph*: We put an edge on the data points $\mathbf{x}_i$ and $\mathbf{x}_j$ if $(\mathbf{x}_i, \mathbf{x}_j)$ is among the k_2 shortest pairs of two different classes, i.e., $\mathbf{x}_i \in c_a$ and $\mathbf{x}_j \in c_b$

$$W'_{ij} = \begin{cases} \|\mathbf{x}_i - \mathbf{x}_j\|^2, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are connected in} \\ & \text{the between-class graph.} \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

Note that here we define the weights by the distance of data pairs. Another simple alternative approach is to define the weights by 0 (connected) and 1 (disconnected).

By maximizing the between-class separability defined in (6) and minimizing the within-class similarity in (5), we propose the following objective function:

$$\arg\max_{\mathbf{A}} \frac{Tr(\mathbf{A}^T\mathbf{X}(\mathbf{D}' - \mathbf{W}')\mathbf{X}^T\mathbf{A})}{Tr(\mathbf{A}^T\mathbf{X}(\mathbf{D} - \mathbf{W})\mathbf{X}^T\mathbf{A})} \quad (10)$$

which can be solved with the following generalized eigendecomposition problem [32]:

$$\mathbf{X}(\mathbf{D}' - \mathbf{W}')\mathbf{X}^T\mathbf{a}_i = \lambda_1\mathbf{X}(\mathbf{D} - \mathbf{W})\mathbf{X}^T\mathbf{a}_i. \quad (11)$$

To be specific, the optimal $\mathbf{A}$ consists of the eigenvectors corresponding to the d largest eigenvalues, i.e.,

$$\mathbf{A} = \{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{d-1}\}. \quad (12)$$

B. Dimension Estimation

As we have proved that the optimal projection matrix $\mathbf{A}$ consists of d eigenvectors in (11). A crucial problem is how to automatically determine the feature dimension d. Most of the existing methods find the dimension d by exhaustively enumerating all possible dimension based on the classification accuracy, which is ad hoc and computationally extensive. The proposed method MMFA can automatically estimate the dimension with the rank of the between-class matrix $\mathbf{W}'$ using the following theorem.

Theorem 1: For a given data set $\mathbf{X}$, the feature dimension d can be estimated by the rank of $\mathbf{D}' - \mathbf{W}'$, i.e.,

$$n - k_2 \times n_c \leq d \leq \min(m, n). \quad (13)$$

Theorem 1 helps determine the feature dimensions and the feature dimension set to the lower bound in our experiments, i.e., $d = n - k_2 \times n_c$. Due to space limitation, we present the proof in the supplementary material. A detailed algorithm of MMFA is summarized in Algorithm 1.

Algorithm 1: Multiple Marginal Fisher Analysis.

Input: A given data set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^{m\times n}$, the label information $\mathbf{c}$, and the nearest neighbor number k_1 and k_2 of within-class graph and between-class graph.

1: Construct the between-class separability and within-class similarity matrixes:
   1) *Within-class graph:* For each sample $\mathbf{x}_i$, set $W_{ij} = W_{ji} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$ or 1 if $\mathbf{x}_j$ is among the k_1 NNs of $\mathbf{x}_i$ in the same class otherwise 0.
   2) *Between-class graph:* For every two classes c_a and c_b, set $W_{ij} = W_{ji} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$ or 1 if $\mathbf{x}_j$ if the pair (i, j) is among the k_2 shortest pairs among the set$\{(i, j)|\mathbf{x}_i \in c_a, \mathbf{x}_j \in c_b\}$ otherwise 0.
2: Compute the eigenvalues and eigenvectors as

$$\mathbf{X}(\mathbf{D}' - \mathbf{W}')\mathbf{X}^T\mathbf{a}_i = \lambda_i\mathbf{X}(\mathbf{D} - \mathbf{W})\mathbf{X}^T\mathbf{a}_i.$$

Thus, $\mathbf{A} = \{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{d-1}\}$, $d = n - k_2 \times n_c$, and $\mathbf{a}_i$ is the eigenvector corresponding to the ith largest eigenvalue λ_i.

Output: The low dimensional embeddings are obtained by

$$\mathbf{Y} = \mathbf{A}^T\mathbf{X}.$$

C. Discussion

Different from existing automatic dimension reduction methods such as LDA and MFA, MMFA is with provable lower and upper bounder in feature dimension. More specifically, LDA has $n_c - 1$ features at most, whereas MMFA has $n - k_2 \times n_c$ at least. Therefore, the feature learned by LDA will be informatively less than that by our MMFA, especially the dataset is large. Moreover, compared with MFA, MMFA has a smaller parameter selecting range on k_2, which can save much computation time. In MMFA, k_1 and k_2 range between 1 and n_i, where n_i denotes the mean number of samples for each class. In contrast, MFA needs to set k_2 from 1 to n.

D. Computational Complexity Analysis

For a given data set $\mathbf{X} \in \mathbb{R}^{m\times n}$, MMFA constructs the aforementioned graphs in $O\big((k_1 + k_2 * \frac{n_c*(n_c-1)}{2})n^2\big)$. Finally, MMFA performs eigendecomposition on (11) in $O(m^3)$. Thus, the time complexity of MMFA is $O(n_c^2n^2 + m^3)$ due to $k_1, k_2 \ll n_c^2$.

IV. Experiments and Results

In this section, we compare the proposed MMFA with seven state-of-the-art dimension reduction methods including LDA [23], MFA [5], LDE [33], PCE [21], PCA [2], NPE [8], and NMF [34]. The baseline results without any dimension reduction are also provided.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS USING THE AR FACES

S_1/S_2	7/7			5/9			3/11		
Algorithms	Accuracy (%)	Time (s)	Para.	Accuracy (%)	Time (s)	Para.	Accuracy (%)	Time (s)	Para.
Baseline	61.17±2.23	-	-	51.97±1.97	-	-	38.65±1.28	-	-
MMFA1	92.94±1.45	10.08±0.42	3, 6	87.82±1.07	8.15±0.35	2, 4	76.38±0.69	7.09±0.42	1, 2
MMFA2	**93.20±0.85**	9.67±0.61	3, 5	**88.48±2.09**	8.31±0.75	2, 3	**76.56±0.66**	7.23±0.32	1, 2
MFA	92.62±1.00	8.12±0.90	3, 200	87.51±1.60	7.59±0.74	2, 120	74.61±1.02	6.89±0.41	2, 80
LDA	92.74±1.34	8.62±0.47	99	87.62±2.20	6.57±0.58	99	75.72±0.69	5.62±0.43	99
LDE	91.54±1.45	11.58±0.88	1, 40	82.08±1.67	8.76±0.52	1, 20	66.78±0.66	6.70±0.64	1, 70
PCE	87.40±1.89	9.59±0.86	20	80.00±0.69	8.57±0.65	25	66.20±2.62	9.01±0.85	60
PCA	61.37±1.98	8.94±1.18	-	51.75±1.63	9.79±1.09	-	38.83±0.86	10.37±0.50	-
NPE	81.42±1.03	5.30±0.33	98	77.68±1.28	4.02±0.47	110	68.05±1.77	1.31±0.20	55
NMF	61.54±4.49	71.92±0.74	-	50.62±4.38	65.26±0.86	-	29.61±5.42	59.29±0.65	-

Note: All methods except PCE, MMFA, MFA, and LDA extract 300 features for classification. Note that S_1/S_2 denotes S_1 training samples for each subject, S_2 denotes n testing samples for each subject. The significant level is fixed to 0.05.

A. Experiment Settings and Datasets

We carry out experiments on three real-world datasets including AR facial database [35], extended Yale dataset B [36], CASIA-3D FaceV1,[1] and UPSPS dataset.[2] To evaluate the performance of the tested methods, we use the extracted features for classification and accuracy as the performance. The used datasets are as follows.

AR face images: The used AR dataset [21] contains three subsets. One contains 1400 clean faces of 100 subjects with different facial expressions and illuminations. The other two subsets are disguised by sunglasses or scarves, both of them contain 600 samples of 100 subjects. Each image is with the size of 55×40.

Extend Yale B face image: The used dataset [21] contains 2204 samples of 38 subjects (58 samples each) and all images are cropped to the size of 54×48.

CASIA-3D FaceV1: The dataset contains 4624 samples of 123 subjects under different illumination, expression, and poses. In the experiment, we use all front faces which contain 1000 images from 100 subjects (10 samples each). All the images are with the size of 60×50.

USPS digits: The dataset contains 11 000 samples of 10 digits ($0 \sim 9$). All the images are with the size of 16×16.

Like [21], we employ the NN classifier to investigate the performance of these feature extraction methods in terms of classification accuracy and time cost. Note that, MMFA, LDA, MFA, and PCE can automatically estimate the feature dimension with different values. We set $d = n - k_2 \times n_c$ in MMFA as described in Section III. Following the experiment settings in [21], all non-adaptive methods reduce the dimension to 300. Like in [5] and [23], we first perform PCA on the input data to preserve $n - n_c$ dimension to avoid the singular problem before MMFA, MFA, and LDA. In experiments, we report the best results by exploring some possible parameter values. More specifically, we set k_1 and k_2 of MMFA between 1 and n_i, where n_i denotes the number of samples for each class. Note that we have provided two weight definition choices: 0/1 (connected/disconnected) or the distance of connected pairs. In the following experiments, we use MMFA1 to denote the first method and MMFA2 denotes the latter one.

For all the evaluated methods, we report the mean and standard deviation of classification accuracy over five randomly sampling data partitions.

B. Performance on Clean Data

In this section, we report the experimental results on the clean datasets including AR, extend Yale B, and CAISA. In order to investigate the influence of different ratio between training and testing size, we randomly split each dataset into two parts with different training–testing ratio. The training/testing data size is denoted by S_1/S_2, where S_1 denotes the S_1 samples of each subject in training data and S_2 denotes the n samples of each subject in testing data.

In the experiments, we employ the NN classifier to evaluate the dimension reduction performance. Both the classification accuracy and time costs are reported in Tables I–III from which one could observe the following.

1) In most cases, MMFA remarkably outperforms the other methods on the three datasets with the NN classifier.
2) For the different training and testing size, MMFA outperforms the baselines on AR and CASIA. On extend Yale B, MMFA obtains better results in the case of 29/29 and 10/48, and is competitive to LDE.
3) Though MMFA considers the multiple marginal pairs, the computation time increases a little as one could see in the tables.

C. Performance on Corrupted and Disguised Images

In this section, we evaluate the robustness of MMFA against corrupted and disguised images, which are as follows.

[1]CASIA-3D FaceV1, http://biometrics.idealtest.org/
[2]USPS, http://archive.ics.uci.edu/ml/datasets.html

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS USING THE EXTEND YALE B FACES

S_1/S_2	29/29			15/43			10/48		
Algorithms	Accuracy (%)	Time (s)	Para.	Accuracy (%)	Time (s)	Para.	Accuracy (%)	Time (s)	Para.
Baseline	66.64±1.46	-	-	54.44±1.32	-	-	45.95±0.88	-	-
MMFA1	**98.38±0.44**	21.49±1.30	18, 10	95.50±0.52	13.07±0.55	5, 5	91.77±0.35	12.18±0.51	3, 3
MMFA2	98.11±0.47	22.81±1.12	22, 9	95.45±0.45	12.89±0.83	6, 4	**92.09±0.62**	11.18±0.62	3, 4
MFA	76.27±1.80	14.78±1.13	1, 780	82.09±0.67	11.81±0.69	1, 740	91.75±0.30	11.52±0.53	3, 780
LDA	97.96±0.35	27.84±2.18	37	94.82±0.37	11.86±1.08	37	90.00±0.73	10.12±0.40	37
LDE	98.16±0.35	22.68±0.69	8, 100	**95.55±0.42**	14.38±1.14	3, 10	91.39±0.34	12.84±1.19	2, 80
PCE	96.33±0.38	15.04±1.04	15	93.28±0.76	14.06±0.65	35	89.52±0.72	13.94±1.00	75
PCA	77.36±1.34	14.90±1.12	-	63.53±0.68	13.32±1.13	-	53.27±1.44	16.14±2.40	-
NPE	89.78±1.29	20.07±0.77	288	89.64±0.65	2.39±0.10	30	87.97±0.83	1.19±0.17	20
NMF	83.68±2.28	88.71±0.63	-	73.24±0.89	73.19±0.37	-	58.49±2.67	67.91±1.00	-

Note: The significant level is fixed to 0.05.

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS USING THE CASIA 3DV1 FACES

S_1/S_2	5/5			4/6			3/7		
Algorithms	Accuracy (%)	Time (s)	Para.	Accuracy (%)	Time (s)	Para.	Accuracy (%)	Time (s)	Para.
Baseline	83.63±2.17	-	-	78.66±1.62	-	-	71.71±1.50	-	-
MMFA1	91.40±1.70	20.05±2.05	1, 1	**90.43±1.30**	21.87±2.84	1, 2	**85.08±0.87**	18.74±1.47	1, 1
MMFA2	**91.44±2.18**	21.23±1.87	1, 1	90.36±1.15	22.45±2.33	1, 2	84.59±1.21	19.12±1.68	1, 1
MFA	90.24±1.47	17.49±1.14	2, 120	88.66±1.80	17.91±1.66	2, 180	83.48±1.11	18.08±1.50	1, 140
LDA	90.99±1.27	16.71±0.49	99	89.53±1.71	16.96±1.58	99	83.97±1.26	17.24±0.89	99
LDE	90.84±1.12	23.15±3.79	1, 50	86.16±1.20	21.66±3.00	1, 30	80.85±1.76	20.61±3.43	1, 90
PCE	90.42±1.19	24.83±2.12	20	88.03±1.96	23.41±2.05	15	84.17±2.89	21.69±2.42	60
PCA	91.08±1.68	18.13±0.91	-	89.20±1.84	19.58±1.55	-	84.08±2.41	16.86±0.43	-
NPE	91.24±1.61	4.07±0.35	95	89.16±1.63	3.63±0.30	95	84.94±2.52	2.93±0.29	115
NMF	74.40±3.69	80.24±0.43	-	68.16±7.44	77.85±0.49	-	42.42±6.53	73.70±0.71	-

Note: The significant level is fixed to 0.05.

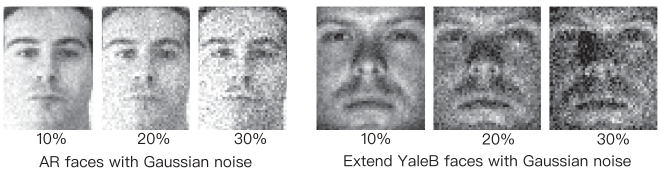

Fig. 4. Some samples from AR faces and extended Yale B faces with Gaussian noise, where the noise ratio increases from 10 to 30%.

1) Corrupted Data: First, we investigate the performance of MMFA on the AR faces and extend Yale with Gaussian noise which is the most common-seeing noise in real world. The Gaussian noise is added via $\mathbf{x}_i' = \mathbf{x}_i + \rho\mathbf{n}$, where ρ is the noise ratio, and $\mathbf{n}$ denotes the noise following the Gaussian distribution. Fig. 4 shows some sample images with the corruption. In this experiment, we only randomly add Gaussian noise into a half of faces, namely, half of the faces are clean and half of them are corrupted. Similar to the experiments on the clean data, we evaluate the performance of different training/testing size using the NN classifier.

Both the mean and the standard deviation of classification accuracy are reported in Tables IV and V, from which we can see that MMFA is more robust than other methods in the most experiments on AR and extend Yale data.

2) Disguised Data: In practice, a large area of images may be corrupted as shown in Fig. 5. In this section, we conduct two experiments with such a case by using disguised AR images. The first experiment is carried out on the AR faces disguised by scarves (occlusion rate is about 40%). The used dataset contains 600 clean samples and 600 disguised samples. The second test is conducted on AR faces disguised by sunglasses (occlusion rate is about 20%), where the dataset contains 600 clean samples and 600 disguised samples. In these two experiments, we randomly generated five different data partitions and each partition contains training and testing subsets with equal size. From Tables VI and VII, one can observe that MMFA outperforms all the baselines on these two disguises.

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS USING THE AR FACES CORRUPTED BY GAUSSIAN NOISE

Gaussian ratio	10%			20%			30%		
Algorithms	Accuracy (%)	Time (s)	Para.	Accuracy (%)	Time (s)	Para.	Accuracy (%)	Time (s)	Para.
Baseline	37.71±1.19	-	-	36.57±1.14	-	-	34.60±1.29	-	-
MMFA1	**89.51±1.60**	10.92±1.48	3, 6	85.60±0.75	12.23±0.87	3, 6	**83.17±1.87**	11.54±1.38	3, 6
MMFA2	89.48±1.86	11.31±1.32	3, 5	**86.20±1.36**	12.42±0.94	3, 5	82.91±1.67	12.03±1.84	3, 5
MFA	88.85±1.30	7.68±0.31	3, 260	85.34±0.64	9.66±0.31	3, 120	82.08±2.46	7.34±0.45	3, 200
LDA	89.28±1.41	9.48±0.57	99	85.22±0.42	12.20±0.38	99	82.71±1.33	9.42±0.57	99
LDE	88.37±1.55	10.81±0.75	1, 20	81.62±2.54	13.39±0.59	1, 40	77.71±0.42	10.10±0.74	1, 40
PCE	86.20±1.27	8.61±0.61	20	84.60±1.75	11.16±0.60	20	80.09±1.29	8.84±0.39	10
PCA	58.62±2.22	9.39±1.23	-	53.28±1.60	8.31±0.87	-	50.48±1.18	9.48±1.30	-
NPE	77,37±2.25	6.15±0.54	110	71.65±1.04	8.01±0.91	115	65.99±1.91	6.75±0.45	115
NMF	54.08±3.37	69.98±0.21	-	54.80±3.34	73.52±0.44	-	53.74±1.79	70.36±0.36	-

Note: All methods except PCE, MMFA, MFA, and LDA extract 300 features for classification. The significant level is fixed to 0.05.

TABLE V
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS USING THE EXTEND YALE B FACES CORRUPTED BY GAUSSIAN NOISE

Gaussian ratio	10%			20%			30%		
Algorithms	Accuracy (%)	Time (s)	Para.	Accuracy (%)	Time (s)	Para.	Accuracy (%)	Time (s)	Para.
Baseline	67.78±1.09	-	-	64.24±1.15	-	-	56.86±1.06	-	-
MMFA1	95.91±0.30	19.87±1.24	6, 7	**94.30±0.68**	24.81±1.15	4, 14	**93.08±0.75**	21.22±0.89	4, 13
MMFA2	95.89±0.59	20.42±1.74	5, 9	93.66±0.87	25.64±1.75	3, 15	92.03±0.79	21.64±0.59	3, 13
MFA	73.72±2.55	17.28±1.70	1, 780	50.19±3.13	17.20±0.53	3, 760	66.51±0.41	14.62±1.17	1, 760
LDA	95.29±0.64	27.37±1.48	37	92.08±0.46	42.42±2.77	37	90.50±0.29	27.79±3.05	37
LDE	**96.17±0.32**	24.96±1.98	3, 10	93.92±1.16	29.67±0.81	4, 20	93.03±0.64	23.92±0.45	5, 40
PCE	95.10±0.50	14.48±0.66	10	94.19±0.71	18.12±1.36	5	92.83±0.74	14.52±1.80	5
PCA	77.16±1.04	14.77±0.99	-	74.82±1.21	13.82±1.70	-	69.56±1.21	14.13±1.78	-
NPE	92.86±0.56	10.21±0.38	125	90.12±0.80	13.44±0.99	115	90.19±0.20	10.57±0.72	120
NMF	67.42±1.41	89.25±0.36	-	81.37±1.22	93.89±0.35	-	57.21±1.38	89.08±0.69	300

Note: The significant level is fixed to 0.05.

Fig. 5. Disguised AR by sunglasses and scarves.

D. Scalability Evaluation

In this section, we give the scalability analysis of MMFA on USPS dataset. In this experiment, we randomly split the dataset into two parts for training and testing, where the number of training samples increases from 500 to 9500 with interval 500. We also compared the other two methods LDA and MFA. The accuracy results are shown in Fig. 6(a). As we can see, the classification accuracy almost remains unchanged when provided 2500 training samples. The computational complexity is $O(n_c^2 n^2 + m^3)$ (Section III-D), which is consistent with the experiment results. To be exact, the increment of computation cost is mostly due to the cost for graph construction, while the embedding cost remains unchanged.

E. Evaluation on Different Classifiers

In order to show the effectiveness of MMFA, we investigate the results of MMFA by using different classifiers compared to other methods. Here, we use three classifiers including NN, support vector machine (SVM), and multilayer perceptron (MLP). Similar to the previous experiments, we randomly split the dataset into two parts with the same size for training and testing. Table VIII shows the results on AR data. As we can see, MMFA outperforms other methods on all the three classifiers.

F. Compared to the Deep Neural Networks

In addition, to show the superiority of our methods, we also compared MMFA to VGG19 network [37], which is pretrained by ImageNet. In the following experiment, we first give the classification accuracy on extracted features obtained by VGG19.

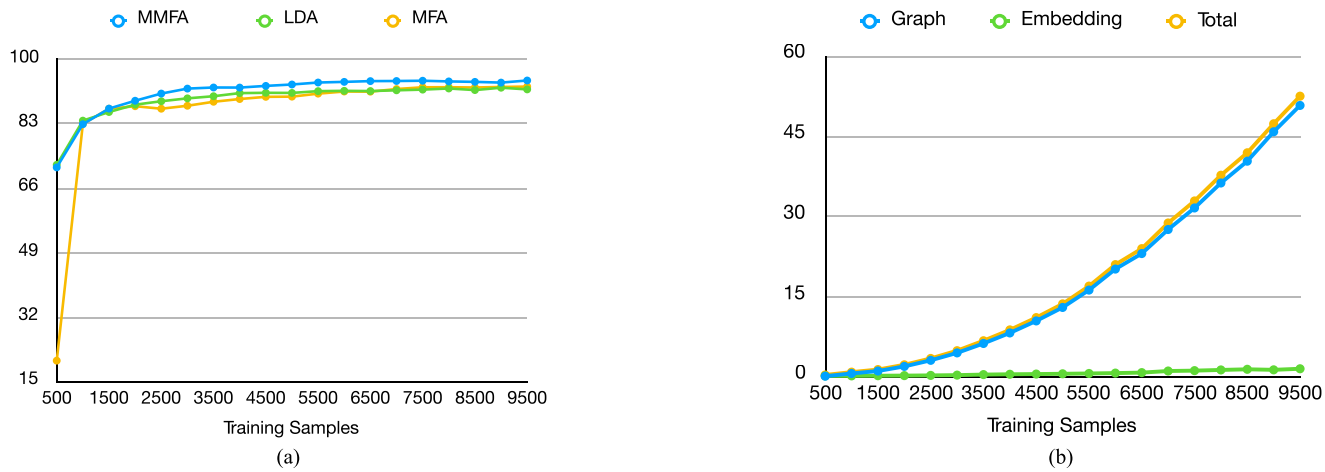

Fig. 6. Scalability analysis of MMFA on the whole USPS dataset, where the training samples increase from 500 to 9500. (a) Classification accuracy of MMFA compared to LDA and MFA. (b) Computation cost of MMFA. Note that the graph time denotes the graph construction cost, and the embedding time denotes the eigendecomposition cost.

TABLE VI
PERFORMANCE ON THE AR DISGUISED BY SCARVES OF DIFFERENT DIMENSION REDUCTION METHODS BASED ON THE NN CLASSIFIER ACCURACY

Algorithms	Accuracy (%)	Time (s)	Para.
Baseline	26.56±0.85	-	-
MMFA1	83.63±2.11	8.62±0.31	3, 5
MMFA2	**84.90±0.94**	9.11±0.59	3, 5
MFA	82.93±2.04	7.43±0.62	3, 120
LDA	83.53±1.73	8.77±1.16	99
LDE	76.46±0.33	9.03±0.53	1, 10
PCE	68.58±1.96	7.84±0.90	55
PCA	26.40±1.29	9.14±1.36	-
NPE	59.40±4.95	8.32±0.28	220
NMF	40.80±2.59	67.59±0.29	-

Note: The time cost for dimension estimation is also investigated. The significant level is fixed to 0.05.

TABLE VII
PERFORMANCE ON THE AR DISGUISED BY SUNGLASSES OF DIFFERENT DIMENSION REDUCTION METHODS BASED ON THE NN CLASSIFIER ACCURACY

Algorithms	Accuracy (%)	Time (s)	Para.
Baseline	33.66±1.47	-	-
MMFA1	86.40±1.12	9.46±0.32	3, 5
MMFA2	**86.56±1.26**	9.26±0.69	2, 4
MFA	85.73±1.10	8.19±1.65	1, 140
LDA	86.26±0.98	9.93±0.66	99
LDE	80.23±1.6	8.89±0.44	1, 10
PCE	66.92±1.75	8.39±1.01	50
PCA	33.56±1.04	8.51±1.72	-
NPE	59.40±1.92	4.48±0.52	65
NMF	39.29±3.55	67.24±0.64	-

Note: The time cost for dimension estimation is also investigated. The significant level is fixed to 0.05.

Furthermore, we also fine-tune the VGG19 model by adding two full-connection layers to obtain classification results in an end-to-end manner. Note that we retrained the VGG19+fine-tune networks on the training data (i.e., AR, Yale B, and CASIA). We randomly split the dataset into two parts with the same size for training and testing. Table IX shows the results from the VGG19 and VGG19+fine-tune. As we can see, both MMFA1 and MMFA2 outperform VGG19 and VGG19+fine-tune.

G. Influence of Parameters

In this section, we investigate the influence of parameters k_1 and k_2 of MMFA. Besides the parameters of MMFA, we also report the performance with varying k in the k-NN classifier. MMFA characterizes the similarity within-class using k_1 neighbors from the same class, while characterizing the separability using k_2 shortest marginal pairs among every two classes. In the experiment, we conduct the experiment on the extend Yale B dataset, which is randomly divided into two parts with equal size for training and testing. In other words, the training data contains 1102 samples over 38 subjects (29 samples each). The evaluation setting is as follows.

1) *Influence of k in k-NN*: We investigate the influence of k (the k-NN classifier) which ranges from 1 to 28 with fixed $k_1 = 5$ and $k_2 = 5$.
2) *Influence of k_1*: As the training data consist of 29 samples for each subject, we fix $k_2 = 5$ and increase k_1 from 1 to 28 according the graph construction strategy.
3) *Influence of k_2*: Similar to k_1, we investigate the performance of MMFA by increasing k_2 from 1 to 28 and fixing $k_1 = 5$.

Note that, we fix k_1 or k_2 to 5 in the above experiment for simplicity. Such a value is not optimal for MMFA.

Fig. 7 shows the influence of parameters. Specifically, Fig. 7(a) shows the performance on the k-NN classifier with different k. Clearly, MMFA first achieves a competitive result and then becomes worse when k increases from 2 to 4. After that,

TABLE VIII
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS WITH DIFFERENT CLASSIFIERS USING AR FACES

S_1/S_2	NN			MLP			SVM		
Algorithms	Accuracy (%)	Time (s)	Para.	Accuracy (%)	Time (s)	Para.	Accuracy (%)	Time (s)	Para.
MMFA1	92.94±1.45	10.08±0.42	3, 6	**90.88±0.70**	11.22±0.52	2, 4	**96.42±0.37**	11.13±0.58	1, 2
MMFA2	**93.20±0.85**	9.67±0.61	3, 6	88.77±1.68	10.27±0.20	4, 5	95.71±0.64	11.68±0.91	1, 2
MFA	92.62±1.00	8.12±0.90	3, 200	90.62±1.55	7.32±0.39	6, 320	95.37±0.92	7.92±0.43	3, 100
LDA	92.74±1.34	8.62±0.47	99	89.17±0.87	8.51±0.66	99	94.65±0.89	8.81+0.39	99
LDE	91.54±1.45	11.58±0.88	1, 40	90.08±0.75	10.32±0.69	3, 50	95.94±0.71	10.79±0.93	1, 40
PCE	87.40±1.89	9.59±0.86	20	88.37±1.20	10.09±0.43	95	94.14±1.07	10.22±0.39	40
PCA	61.37±1.98	8.94±1.18	-	37.02±2.47	7.21±0.78	-	95.42±0.43	6.95±0.86	-
NPE	81.42±1.03	5.30±0.33	98	90.71±0.84	4.87+0.72	140	93.99±0.52	5.19+0.53	140
NMF	61.54±4.49	71.92±0.74	-	88.31±3.19	68.87±1.74	-	90.82±2.5	69.31±0.96	-

Note: The time reported here only includes the dimension reduction cost. The significant level is fixed to 0.05.

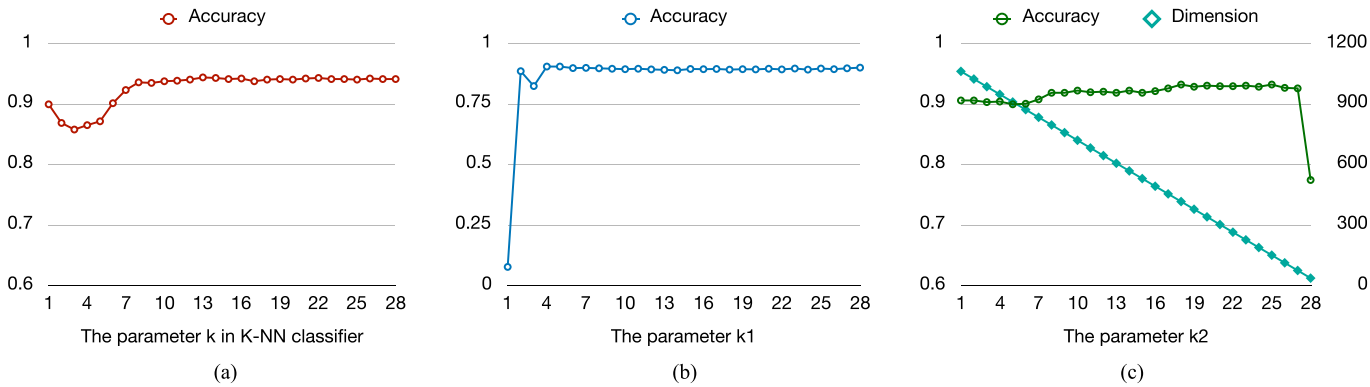

Fig. 7. (a) Classification accuracy with varying parameter k in *k*-NN classifier with $k_1 = 5$ and $k_2 = 5$. (b) Classification accuracy with varying parameter k_1 from 1 to 28 by fixing $k_2 = 5$. (c) Classification accuracy with varying parameter k_1 from 2 to 28 by fixing $k_1 = 5$. In addition, we also show the feature dimension with varying k_2. Note that we use *k*-NN classifier in the experiment (a), while the NN classifier is used in the experiment (b) and (c).

TABLE IX
COMPARISON WITH VGG19 NETWORKS

Methods	AR	Yale	CASIA
MMFA1	**92.94±1.45**	**98.38±0.44**	91.40±1.70
MMFA2	92.83±1.26	98.11±0.47	**91.44±2.18**
VGG19	80.14±1.68	57.96±0.94	82.28±0.89
VGG19+fine-tune	85.77±1.49	66.63±1.97	73.39±2.56

Note: The significant level is fixed to 0.05.

the classification accuracy gradually increases when k increases from 4 to 13. In general, MMFA is robust to varying number k in k-NN classifier, whose classification performance almost keeps unchanged in the case of $k > 8$. Fig. 7(b) and (c) show the influence of k_1 and k_2, respectively. As one can see that, the accuracy of MMFA remarkably increases with k_1, and then gives a slight change when k_1 increases to 4. Regarding to k_2, the accuracy of MMFA increases slowly with k_2, and a decline took place when $k_2 = 28$. We find an interesting observation that the accuracy first increases greatly and remains unchanged at k_1, while the accuracy first increases slowly and decreases greatly at last. The former phenomenon should attribute to that $k_1 = 1$ misses a lot of within-class information, and the latter one may be resulted from that $k_2 = 28$ cannot keep sufficient information to separate heterogeneous data.

V. CONCLUSION

In this paper, we proposed a novel supervised subspace learning method called *MMFA*. Unlike the most existing methods, MMFA can automatically estimate the feature dimension and obtain the low-dimensional representation. Extensive experimental investigations showed that our method could achieve the state of the arts in feature extraction for classifying clean, noisy, and disguised images.

ACKNOWLEDGMENT

The authors would like to thank the associate editor and anonymous reviewers for their valuable comments and constructive suggestions to improve the quality of this paper.

REFERENCES

[1] I. K. Fodor, "A survey of dimension reduction techniques," Center for Applied Scientific Computing, Lawrence Livermore National Laboratory 9, pp. 1–18, 2002.

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

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

[4] X. Peng, Z. Yi, and H. Tang, "Robust subspace clustering via thresholding ridge regression," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 3827–3833.

[5] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[6] M. F. D'Angelo, R. M. Palhares, M. C. C. Filho, R. D. Maia, J. B. Mendes, and P. Y. Ekel, "A new fault classification approach applied to Tennessee Eastman benchmark process," *Appl. Soft Comput.*, vol. 49, pp. 676–686, 2016.

[7] J. B. Tenenbaum, V. D. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[8] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, pp. 1208–1213, vol. 2, 2005.

[9] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[10] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 153–160, 2004.

[11] Z. Kang, C. Peng, and Q. Cheng, "Kernel-driven similarity learning," *Neurocomputing*, vol. 267, pp. 210–219, 2017.

[12] Z. Wang, Y. Yang, S. Chang, Q. Ling, and T. S. Huang, "Learning a deep l_∞ encoder for hashing," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2174–2180.

[13] Z. Kang, C. Peng, Q. Cheng, and Z. Xu, "Unified spectral clustering with optimal graph," 2017, arXiv:1711.04258.

[14] Y. Yuan, J. Wan, and Q. Wang, "Congested scene classification via efficient unsupervised feature learning and density estimation," *Pattern Recognit.*, vol. 56, pp. 159–169, 2016.

[15] Z. Wang, N. M. Nasrabadi, and T. S. Huang, "Semisupervised hyperspectral classification using task-driven dictionary learning with Laplacian regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1161–1173, Mar. 2015.

[16] Q. Wang, Z. Qin, F. Nie, and X. Li, "Spectral embedded adaptive neighbors clustering," *IEEE Trans Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2018.2861209.

[17] L. Qiao, S. Chen, and X. Tan, "Sparsity preserving projections with applications to face recognition," *Pattern Recognit.*, vol. 43, no. 1, pp. 331–341, 2010.

[18] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang, "Learning with L1-graph for image analysis," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 858–866, Apr. 2010.

[19] X. Peng, C. Lu, Z. Yi, and H. Tang, "Connections between nuclear-norm and Frobenius-norm-based representations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 218–224, Jan. 2018.

[20] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the l2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, Apr. 2017.

[21] X. Peng, J. Lu, Z. Yi, and R. Yan, "Automatic subspace learning via principal coefficients embedding," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3583–3596, Nov. 2017.

[22] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Human Genetics*, vol. 7, no. 2, pp. 179–188, 1936.

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

[24] J. Pylkkönen, "LDA based feature estimation methods for LVCSR," in *Proc. 9th Int. Conf. Spoken Lang. Process.*, 2006, pp: 389–392.

[25] Q. Wang, J. Wan, and Y. Yuan, "Locality constraint distance metric learning for traffic congestion detection," *Pattern Recognit.*, vol. 75, pp. 272–281, 2018.

[26] C. Deng, Z. Chen, X. Liu, X. Gao, and D. Tao, "Triplet-based deep hashing network for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3893–3903, Aug. 2018.

[27] X. Li, J. Lv, and Z. Yi, "An efficient representation-based method for boundary point and outlier detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 51–62, Jan. 2018.

[28] Q. Wang, X. He, and X. Li, "Locality and structure regularized low rank representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2018.2862899.

[29] S. Siena, V. N. Boddeti, and V. Kumar, "Coupled marginal Fisher analysis for low-resolution face recognition," in *Proc. 12th Eur. Conf. Comput. Vis.*, Springer, 2012, pp. 240–249.

[30] R. A. Fisher, "The statistical utilization of multiple measurements," *Ann. Human Genetics*, vol. 8, no. 4, pp. 376–386, 1938.

[31] Q. Wang, F. Zhang, and X. Li, "Optimal clustering framework for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, to be published. doi: 10.1109/TGRS.2018.2828161.

[32] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York, NY, USA: Academic, 2013.

[33] H.-T. Chen, H.-W. Chang, and T.-L. Liu, "Local discriminant embedding and its variants," in *Proc. 18th IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 2, pp. 846–853.

[34] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, 2004.

[35] A. Martinez and R. Benavente, "The AR face database," Ohio State Univ., Columbus, OH, CVC Tech. Rep. 24, Jun. 1998.

[36] A. S. Georghiades, P. N. Belhumeur, and D. J. 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, Jun. 2001.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, arXiv:1409.1556.

Zhenyu Huang received the bachelor's degree in computer science from Sichuan University, Chengdu, China, in 2018. He is currently working toward the master's degree in computer science with the College of Computer Science, Sichuan University, Chengdu, China.

His research interests include deep learning and clustering.

Hongyuan Zhu received the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore, in 2014.

He is currently a Research Scientist with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. His research interests include multimedia content analysis and segmentation, especially image segmentation/cosegmentation, object detection, scene recognition, and saliency detection.

Joey Tianyi Zhou received the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2015.

He is a Scientist with the Institute of High Performance Computing, Research Agency for Science, Technology and Research, Singapore.

Dr. Zhou was the recipient of the NIPS 2017 Best Reviewer Award, the Best Paper Nomination at ECCV 2016, and the Best Poster Honorable Mention at ACML 2012.

Xi Peng (S'09–M'14) received the Ph.D. degree in computer science from Sichuan University, Chengdu, China, in 2013.

He is a Research Professor with the College of Computer Science, Sichuan University, Chengdu, China.

Prof. Peng has served as an Associate/Guest Editor for several journals including IEEE ACCESS and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He has been a Session Chair for AAAI'17 and IJCAI'18, a Senior Program Committee Member for IJCAI'17, and a Chair to organize a tutorial at ECCV'16 and a special session at VCIP'17.