

Automatic Subspace Learning via Principal Coefficients Embedding

Xi Peng, Jiwen Lu, *Senior Member, IEEE*, Zhang Yi, *Fellow, IEEE*, and Rui Yan, *Member, IEEE*

Abstract—In this paper, we address two challenging problems in unsupervised subspace learning: 1) how to automatically identify the feature dimension of the learned subspace (i.e., automatic subspace learning) and 2) how to learn the underlying subspace in the presence of Gaussian noise (i.e., robust subspace learning). We show that these two problems can be simultaneously solved by proposing a new method [(called principal coefficients embedding (PCE))]. For a given data set $\mathbf{D} \in \mathbb{R}^{m \times n}$, PCE recovers a clean data set $\mathbf{D}_0 \in \mathbb{R}^{m \times n}$ from $\mathbf{D}$ and simultaneously learns a global reconstruction relation $\mathbf{C} \in \mathbb{R}^{n \times n}$ of $\mathbf{D}_0$. By preserving $\mathbf{C}$ into an m' -dimensional space, the proposed method obtains a projection matrix that can capture the latent manifold structure of $\mathbf{D}_0$, where $m' \ll m$ is automatically determined by the rank of $\mathbf{C}$ with theoretical guarantees. PCE has three advantages: 1) it can automatically determine the feature dimension even though data are sampled from a union of multiple linear subspaces in presence of the Gaussian noise; 2) although the objective function of PCE only considers the Gaussian noise, experimental results show that it is robust to the non-Gaussian noise (e.g., random pixel corruption) and real disguises; and 3) our method has a closed-form solution and can be calculated very fast. Extensive experimental results show the superiority of PCE on a range of databases with respect to the classification accuracy, robustness, and efficiency.

Index Terms—Automatic dimension reduction, corrupted data, graph embedding, manifold learning, robustness.

I. INTRODUCTION

SUBSPACE learning or metric learning aims to find a projection matrix $\Theta \in \mathbb{R}^{m \times m'}$ from the training data $\mathbf{D}^{m \times n}$, so that the high-dimensional datum $\mathbf{y} \in \mathbb{R}^m$ can be transformed into a low-dimensional space via $\mathbf{z} = \Theta^T \mathbf{y}$. Existing subspace learning methods can be roughly divided into three categories: supervised, semi-supervised, and unsupervised. Supervised method incorporates the class label information of $\mathbf{D}$ to obtain

discriminative features. The well-known works include but not limit to linear discriminant analysis [1], neighborhood components analysis [2], and their variants such as [3]–[6]. Moreover, Xu *et al.* [7] recently proposed to formulate the problem of supervised multiple view subspace learning as one multiple source communication system, which provide a novel insight to the community. Semi-supervised methods [8]–[10] utilize limited labeled training data as well as unlabeled ones for better performance. Unsupervised methods seek a low-dimensional subspace without using any label information of training samples. Typical methods in this category include Eigenfaces [11], neighborhood preserving embedding (NPE) [12], locality preserving projections (LPPs) [13], sparsity preserving projections (SPPs) [14] or known as L1-graph [15], and multiview intact space learning [16]. For these subspace learning methods, Yan *et al.* [17] have shown that most of them can be unified into the framework of graph embedding, i.e., low dimensional features can be achieved by embedding some desirable properties (described by a similarity graph) from a high-dimensional space into a low-dimensional one. By following this framework, this paper focuses on unsupervised subspace learning, i.e., dimension reduction considering the unavailable label information in training data.

Although a large number of subspace learning methods have been proposed, less works have investigated the following two challenging problems simultaneously: 1) how to automatically determine the dimension of the feature space, referred to as automatic subspace learning and 2) how to immune the influence of corruptions, referred to as robust subspace learning.

Automatic subspace learning involves the technique of dimension estimation which aims at identifying the number of features necessary for the learned low-dimensional subspace to describe a data set. In previous studies, most existing methods manually set the feature dimension by exploring all possible values based on the classification accuracy. Clearly, such a strategy is time-consuming and easily overfits to the specific data set. In the literature of manifold learning, some dimension estimation methods have been proposed, e.g., spectrum analysis based methods [18], [19], box-counting based methods [20], fractal-based methods [21], [22], tensor voting [23], and neighborhood smoothing [24]. Although these methods have achieved some impressive results, this problem is still far from solved due to the following limitations: 1) these existing methods may work only when data are sampled in a uniform way and data are free to corruptions, as pointed

Manuscript received August 20, 2015; revised December 2, 2015 and February 24, 2016; accepted May 22, 2016. Date of publication June 9, 2016; date of current version October 13, 2017. This work was supported by in part by A*STAR Industrial Robotics Program—Distributed Sensing and Perception under SERC Grant 1225100002, and in part by the National Natural Science Foundation of China under Grant 61432012. This paper was recommended by Associate Editor D. Tao. (*Corresponding author: Rui Yan.*)

X. Peng is with the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore 138632 (e-mail: pangsaai@gmail.com).

J. Lu is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: lujiwen@mails.tsinghua.edu.cn).

Z. Yi and R. Yan are with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: zhangyi@scu.edu.cn; yanrui2006@gmail.com).

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/TCYB.2016.2572306

out by Saul and Roweis [25]; 2) most of them can accurately estimate the intrinsic dimension of a single subspace but would fail to work well for the scenarios of multiple subspaces, especially, in the case of the dependent or disjoint subspaces; and 3) although some dimension estimation techniques can be incorporated with subspace learning algorithms, it is more desirable to design a method that can not only automatically identify the feature dimension but also reduce the dimension of data.

Robust subspace learning aims at identifying underlying subspaces even though the training data $\mathbf{D}$ contains gross corruptions such as the Gaussian noise. Since $\mathbf{D}$ is corrupted by itself, accurate prior knowledge about the desired geometric properties is hard to be learned from $\mathbf{D}$. Furthermore, gross corruptions will make dimension estimation more difficult. This so-called robustness learning problem has been challenging in machine learning and computer vision. One of the most popular solutions is recovering a clean data set from inputs and then performing dimension reduction over the clean data. Typical methods include the well-known principal component analysis (PCA) which achieves robust results by removing the bottom eigenvectors corresponding to the smallest eigenvalues. However, PCA can achieve a good result only when data are sampled from a single subspace and only contaminated by a small amount of noises. Moreover, PCA needs specifying a parameter (e.g., 98% energy) to distinct the principal components from the minor ones. To improve the robustness of PCA, Candès *et al.* [26] recently proposed robust PCA (RPCA) which can handle the sparse corruption and has achieved a lot of success [27]–[32]. However, RPCA directly removes the errors from the input space, which cannot obtain the low-dimensional features of inputs. Moreover, the computational complexity of RPCA is too high to handle the large-scale data set with very high dimensionality. Bao *et al.* [33] proposed an algorithm which can handle the gross corruption. However, they did not explore the possibility to automatically determine feature dimension. Tzimiropoulos *et al.* [34] proposed a subspace learning method from image gradient orientations by replacing pixel intensities of images with gradient orientations. Shu *et al.* [35] proposed to impose the low-rank constraint and group sparsity on the reconstruction coefficients under orthonormal subspace so that the Laplacian noise can be identified. Their method outperforms a lot of popular methods such as Gabor features in illumination- and occlusion-robust face recognition. Liu and Tao [36] have recently carried out a series of comprehensive works to discuss how to handle various noises, e.g., Cauchy noise, Laplacian noise [37], and noisy labels [38]. Their works provide some novel theoretical explanations toward understanding the role of these errors. Moreover, some recent developments have been achieved in the field of subspace clustering [39]–[45], which use ℓ_1 -, ℓ_2 -, or nuclear-norm based representation to achieve robustness.

In this paper, we proposed a robust unsupervised subspace learning method which can automatically identify the number of features. The proposed method, referred to as principal coefficients embedding (PCE), formulates the possible corruptions as a term of an objective function so that a clean data set

$\mathbf{D}_0$ and the corresponding reconstruction coefficients $\mathbf{C}$ can be simultaneously learned from the training data $\mathbf{D}$. By embedding $\mathbf{C}$ into an m' -dimensional space, PCE obtains a projection matrix $\Theta^{m \times m'}$, where m' is adaptively determined by the rank of $\mathbf{C}$ with theoretical guarantees.

PCE is motivated by a recent work in subspace clustering [39], [46] and the well-known locally linear embedding (LLE) method [47]. The former motivates us the way to achieve robustness, i.e., the errors such as the Gaussian noise can be mathematically formulated as a term into the objective function and thus the errors can be explicitly removed. The major differences between [39] and PCE are: 1) [39] is proposed for clustering, whereas PCE is for dimension reduction; 2) the objective functions are different. PCE is based on the Frobenius norm instead of the nuclear norm, thus resulting a closed-form solution and avoiding iterative optimization procedure; and 3) PCE can automatically determine the feature dimension, whereas [39] does not investigate this challenging problem. LLE motivated us the way to estimate feature dimension even though it does not overcome this problem. LLE is one of most popular dimension reduction methods, which encodes each data point as a linear combination of its neighborhood and preserves such reconstruction relationship into different projection spaces. LLE implies the possibility to estimate the feature dimension using the size of neighborhood of data points. However, this parameter needs to be specified by users rather than automatically learning from data. Thus, LLE still suffers from the issue of dimension estimation. Moreover, the performance of LLE would be degraded when the data is contaminated by noises. The contributions of this paper are summarized as follows.

- 1) The proposed method (i.e., PCE) can handle the Gaussian noise that probably exists into data with theoretical guarantees. Different from the existing dimension reduction methods such as L1-Graph, PCE formulates the corruption into its objective function and only calculates the reconstruction coefficients using a clean data set instead of the original one. Such a formulation can perform data recovery and improve the robustness of PCE.
- 2) Unlike previous subspace learning methods, PCE can automatically determine the feature dimension of the learned low-dimensional subspace. This largely reduces the efforts for finding an optimal dimension and thus PCE is more competitive in practice.
- 3) PCE is computationally efficient, which only involves performing singular value decomposition (SVD) over the training data set one time.

The rest of this paper is organized as follows. Section II briefly introduces some related works. Section III presents our proposed algorithm. Section IV reports the experimental results and Section V concludes this paper.

II. RELATED WORKS

A. Notations and Definitions

In the following, lower-case bold letters represent column vectors and upper-case bold ones denote matrices. $\mathbf{A}^T$ and

TABLE I
SOME USED NOTATIONS

Notation	Definition
n	the number of data points
n_i	data size of the i -th subject
m	the dimension of input
m'	the dimension of feature space
s	the number of subject
r	the rank of a given matrix
$\mathbf{y} \in \mathbb{R}^m$	a given testing sample
$\mathbf{z} \in \mathbb{R}^{m'}$	the low-dimensional feature
$\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n]$	training data set
$\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^T$	full and skinny SVD of $\mathbf{D}$
$\mathbf{D}_0 \in \mathbb{R}^{m \times n}$	the desired clean data set
$\mathbf{E} \in \mathbb{R}^{m \times n}$	the errors existing into $\mathbf{D}$
$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n]$	the representation of $\mathbf{D}_0$
$\sigma_i(\mathbf{C})$	the i -th singular value of $\mathbf{C}$
$\mathbf{\Theta} \in \mathbb{R}^{m \times m'}$	the projection matrix

$\mathbf{A}^\dagger$ denote the transpose and pseudo-inverse of the matrix $\mathbf{A}$, respectively. $\mathbf{I}$ denotes the identity matrix.

For a given data matrix $\mathbf{D} \in \mathbb{R}^{m \times n}$, the Frobenius norm of $\mathbf{D}$ is defined as

$$\|\mathbf{D}\|_F = \sqrt{\text{trace}(\mathbf{D}\mathbf{D}^T)} = \sqrt{\sum_{i=1}^{\min\{m,n\}} \sigma_i^2(\mathbf{D})} \quad (1)$$

where $\sigma_i(\mathbf{D})$ denotes the i th singular value of $\mathbf{D}$.

The full SVD and the skinny SVD of $\mathbf{D}$ are defined as $\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ and $\mathbf{D} = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^T$, where $\mathbf{\Sigma}$ and $\mathbf{\Sigma}_r$ are in descending order. $\mathbf{U}_r$, $\mathbf{V}_r$ and $\mathbf{\Sigma}_r$ consist of the top (i.e., largest) r singular vectors and singular values of $\mathbf{D}$. Table I summarizes some notations used throughout this paper.

B. Locally Linear Embedding

Yan *et al.* [17] have shown that most unsupervised, semi-supervised, and supervised subspace learning methods can be unified into a framework known as graph embedding. Under this framework, subspace learning methods obtain low-dimensional features by preserving some desirable geometric relationships from a high-dimensional space into a low-dimensional one. Thus, the performance of subspace learning largely depends on the identified relationship which is usually described by a similarity graph (i.e., affinity matrix). In the graph, each vertex corresponds to a data point and the edge weight denotes the similarity between two connected points. There are two popular ways to measure the similarity among data points, i.e., pairwise distance such as Euclidean distance [48] and linear reconstruction coefficients introduced by LLE [47].

For a given data matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n]$, LLE solves the following problem:

$$\min_{\mathbf{c}_i} \sum_{i=1}^n \|\mathbf{d}_i - \mathbf{B}_i \mathbf{c}_i\|_2, \text{ s.t. } \sum_j c_{ij} = 1 \quad (2)$$

where $\mathbf{c}_i \in \mathbb{R}^p$ is the linear representation of $\mathbf{d}_i$ over $\mathbf{B}_i$, c_{ij} denotes the j th entry of $\mathbf{c}_i$, and $\mathbf{B}_i \in \mathbb{R}^{m \times p}$ consists of p nearest neighbors (NNs) of $\mathbf{d}_i$ that are chosen from the collection of $[\mathbf{d}_1, \dots, \mathbf{d}_{i-1}, \mathbf{d}_{i+1}, \dots, \mathbf{d}_n]$ in terms of Euclidean distance.

By assuming the reconstruction relationship $\mathbf{c}_i$ is invariant to ambient space, LLE obtains the low-dimensional features $\mathbf{Y} \in \mathbb{R}^{m' \times n}$ of $\mathbf{D}$ by

$$\min_{\mathbf{Y}} \|\mathbf{Y} - \mathbf{Y}\mathbf{W}\|_F^2, \text{ s.t. } \mathbf{Y}^T \mathbf{Y} = \mathbf{I} \quad (3)$$

where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n]$ and the nonzero entries of $\mathbf{w}_i \in \mathbb{R}^n$ corresponds to $\mathbf{c}_i$.

However, LLE cannot handle the out-of-sample data that are not included into $\mathbf{D}$. To solve this problem, NPE [48] calculates the projection matrix $\mathbf{\Theta}$ instead of $\mathbf{Y}$ by replacing $\mathbf{Y}$ with $\mathbf{\Theta}^T \mathbf{D}$ into (3).

C. L1-Graph

By following the framework of LLE and NPE, Qiao *et al.* [14] and Cheng *et al.* [15] proposed SPP and L1-graph, respectively. The methods sparsely encode each data points by solving the following sparse coding problem:

$$\min_{\mathbf{c}_i} \|\mathbf{d}_i - \mathbf{D}_i \mathbf{c}_i\|_2 + \lambda \|\mathbf{c}_i\|_1 \quad (4)$$

where $\mathbf{D}_i = [\mathbf{d}_1, \dots, \mathbf{d}_{i-1}, \mathbf{0}, \mathbf{d}_{i+1}, \dots, \mathbf{d}_n]$ and (4) can be solved by many ℓ_1 -solvers [49], [50].

After obtaining $\mathbf{C} \in \mathbb{R}^{n \times n}$, SPP and L1-graph embed $\mathbf{C}$ into the feature space by following NPE. The advantage of sparsity based subspace methods is that they can automatically determine the neighborhood for each data point without the parameter of neighborhood size. Inspired by the success of SPP and L1-graph, a number of spectral embedding methods [51]–[56] have been proposed. However, these methods including L1-graph and SPP have still required specifying the dimension of feature space.

D. Robust Principal Component Analysis

RPCA [26] is proposed to improve the robustness of PCA, which solves the following optimization problem:

$$\min_{\mathbf{D}_0, \mathbf{E}} \text{rank}(\mathbf{D}_0) + \lambda \|\mathbf{E}\|_0, \text{ s.t. } \mathbf{D} = \mathbf{D}_0 + \mathbf{E} \quad (5)$$

where $\lambda > 0$ is the parameter to balance the possible corruptions and the desired clean data, and $\|\cdot\|_0$ is ℓ_0 -norm to count the number of nonzero entries of a given matrix or vector.

Since the rank operator and ℓ_0 -norm are nonconvex and discontinuous, ones usually relax them with nuclear norm and ℓ_1 -norm [57]. Then, (5) is approximated by

$$\min_{\mathbf{D}_0, \mathbf{E}} \|\mathbf{D}_0\|_* + \lambda \|\mathbf{E}\|_1, \text{ s.t. } \mathbf{D} = \mathbf{D}_0 + \mathbf{E} \quad (6)$$

where $\|\mathbf{D}\|_* = \text{trace}(\sqrt{\mathbf{D}^T \mathbf{D}}) = \sum_{i=1}^{\min\{m,n\}} \sigma_i(\mathbf{D})$ denotes the nuclear norm of $\mathbf{D}$ and $\sigma_i(\mathbf{D})$ is the i th singular value of $\mathbf{D}$.

III. PRINCIPAL COEFFICIENTS EMBEDDING FOR UNSUPERVISED SUBSPACE LEARNING

In this section, we propose an unsupervised algorithm for subspace learning, i.e., PCE. The method not only can achieve robust results but also can automatically determine the feature dimension.

For a given data set $\mathbf{D}$ containing the errors $\mathbf{E}$, PCE achieves robustness and dimension estimation in two steps.

- 1) The first step achieves the robustness by recovering a clean data set $\mathbf{D}_0$ from $\mathbf{D}$ and building a similarity graph $\mathbf{C} \in \mathbb{R}^{n \times n}$ with the reconstruction coefficients of $\mathbf{D}_0$, where $\mathbf{D}_0$ and $\mathbf{C}$ are jointly learned by solving an SVD problem.
- 2) The second step automatically estimates the feature dimension using the rank of $\mathbf{C}$ and learns the projection matrix $\Theta \in \mathbb{R}^{m \times m'}$ by embedding $\mathbf{C}$ into an m' -dimensional space. In the following, we will introduce these two steps in details.

A. Robustness Learning

For a given training data matrix $\mathbf{D}$, PCE removes the corruption $\mathbf{E}$ from $\mathbf{D}$ and then linearly encodes the recovered clean data set $\mathbf{D}_0$ over itself. The proposed objective function is as follows:

$$\min_{\mathbf{C}, \mathbf{D}_0, \mathbf{E}} \frac{1}{2} \|\mathbf{C}\|_F^2 + \frac{\lambda}{2} \|\mathbf{E}\|_p^2 \text{ s.t. } \underbrace{\mathbf{D} = \mathbf{D}_0 + \mathbf{E}}_{\text{Robustness}}, \underbrace{\mathbf{D}_0 = \mathbf{D}_0 \mathbf{C}}_{\text{self-expression}}. \quad (7)$$

The proposed objective function mainly considers the constraints on the representation $\mathbf{C}$ and the errors $\mathbf{E}$. We enforce Frobenius norm on $\mathbf{C}$ because some recent works have shown that the Frobenius norm based representation is more computationally efficient than the ℓ_1 - and nuclear-norm based representation while achieving competitive performance in face recognition [58] and subspace clustering [41]. Moreover, Frobenius-norm based representation has shared some desirable properties with nuclear-norm based representation as shown in our previous theoretical studies [46], [59].

The term $\mathbf{D}_0 = \mathbf{D}_0 \mathbf{C}$ is motivated by the recent development in subspace clustering [60], [61], which can be further derived from the formulation of LLE [i.e., (3)]. More specifically, ones reconstruct $\mathbf{D}_0$ by itself to obtain this so-called self-expression as the similarity of data set. The major differences between this paper and the existing methods are: 1) the objective functions are different. Our method is based on Frobenius norm instead of ℓ_1 - or nuclear-norm; 2) the methods directly project the original data $\mathbf{D}$ into the space spanned by itself, whereas we simultaneously learn a clean data set $\mathbf{D}_0$ from $\mathbf{D}$ and compute the self-expression of $\mathbf{D}_0$; and 3) PCE is proposed for subspace learning, whereas the methods are proposed for clustering.

By formulating the error $\mathbf{E}$ as a term into our objective function, we can achieve robustness by $\mathbf{D}_0 = \mathbf{D} - \mathbf{E}$. The constraint on $\mathbf{E}$ (i.e., $\|\cdot\|_p$) could be chosen as ℓ_1 -, ℓ_2 -, or $\ell_{2,1}$ -norm. Different choices of $\|\cdot\|_p$ correspond to different types of noises. For example, ℓ_1 -norm is usually used to formulate the Laplacian noise, ℓ_2 -norm is adopted to describe the Gaussian noise, and $\ell_{2,1}$ -norm is used to represent the sample-specified corruption such as outlier [61]. Here, we mainly consider the Gaussian noise which is commonly assumed in signal transmission problem. Thus, we have the following objective function:

$$\min_{\mathbf{C}, \mathbf{D}_0, \mathbf{E}} \frac{1}{2} \|\mathbf{C}\|_F^2 + \frac{\lambda}{2} \|\mathbf{E}\|_F^2 \text{ s.t. } \mathbf{D} = \mathbf{D}_0 + \mathbf{E}, \mathbf{D}_0 = \mathbf{D}_0 \mathbf{C} \quad (8)$$

where $\|\mathbf{E}\|_F$ denotes the error that follows the Gaussian distribution. It is worthy to point out that, although the above formulation only considers the Gaussian noise, our experimental result show that PCE is also robust to other corruptions such as random pixel corruption (nonadditive noise) and real disguises.

To efficiently solve (8), we first consider the case of corruption-free, i.e., $\mathbf{E} = \mathbf{0}$. In such a setting, (8) is simplified as follows:

$$\min_{\mathbf{C}} \|\mathbf{C}\|_F \text{ s.t. } \mathbf{D} = \mathbf{D} \mathbf{C}. \quad (9)$$

Note that, $\mathbf{D}^\dagger \mathbf{D}$ is a feasible solution to $\mathbf{D} = \mathbf{D} \mathbf{C}$, where $\mathbf{D}^\dagger$ denotes the pseudoinverse of $\mathbf{D}$. In [46], an unique minimizer to (9) is given as follows.

Lemma 1: Let $\mathbf{D} = \mathbf{U}_r \Delta_r \mathbf{V}_r^T$ be the skinny SVD of the data matrix $\mathbf{D} \neq \mathbf{0}$. The unique solution to

$$\min \|\mathbf{C}\|_F \text{ s.t. } \mathbf{D} = \mathbf{D} \mathbf{C} \quad (10)$$

is given by $\mathbf{C}^* = \mathbf{V}_r \mathbf{V}_r^T$, where r is the rank of $\mathbf{D}$ and $\mathbf{D}$ is a clean data set without any corruptions.

Proof: Let $\mathbf{D} = \mathbf{U} \Delta \mathbf{V}^T$ be the full SVD of $\mathbf{D}$. The pseudo-inverse of $\mathbf{D}$ is $\mathbf{D}^\dagger = \mathbf{V}_r \Delta_r^{-1} \mathbf{U}_r^T$. Defining $\mathbf{V}_c$ by $\mathbf{V}^T = \begin{bmatrix} \mathbf{V}_r^T \\ \mathbf{V}_c^T \end{bmatrix}$ and $\mathbf{V}_c^T \mathbf{V}_r = \mathbf{0}$. To prove that $\mathbf{C}^* = \mathbf{V}_r \mathbf{V}_r^T$ is the unique solution to (10), two steps are required.

First, we prove that $\mathbf{C}^*$ is the minimizer to (10), i.e., for any $\mathbf{X}$ satisfying $\mathbf{D} = \mathbf{D} \mathbf{X}$, it must hold that $\|\mathbf{X}\|_F \geq \|\mathbf{C}^*\|_F$. Since for any column orthogonal matrix $\mathbf{P}$, it must hold that $\|\mathbf{P} \mathbf{M}\|_F = \|\mathbf{M}\|_F$. Then, we have

$$\begin{aligned} \|\mathbf{X}\|_F &= \left\| \begin{bmatrix} \mathbf{V}_r^T \\ \mathbf{V}_c^T \end{bmatrix} [\mathbf{C}^* + (\mathbf{X} - \mathbf{C}^*)] \right\|_F \\ &= \left\| \begin{bmatrix} \mathbf{V}_r^T \mathbf{C}^* + \mathbf{V}_r^T (\mathbf{X} - \mathbf{C}^*) \\ \mathbf{V}_c^T \mathbf{C}^* + \mathbf{V}_c^T (\mathbf{X} - \mathbf{C}^*) \end{bmatrix} \right\|_F. \end{aligned} \quad (11)$$

As $\mathbf{C}^*$ satisfies $\mathbf{D} = \mathbf{D} \mathbf{C}^*$, then $\mathbf{D}(\mathbf{X} - \mathbf{C}^*) = \mathbf{0}$, i.e., $\mathbf{U}_r \Delta_r \mathbf{V}_r^T (\mathbf{X} - \mathbf{C}^*) = \mathbf{0}$. Since $\mathbf{U}_r \Delta_r \neq \mathbf{0}$, $\mathbf{V}_r^T (\mathbf{X} - \mathbf{C}^*) = \mathbf{0}$. Denote $\mathbf{\Gamma} = \Delta_r^{-1} \mathbf{U}_r^T \mathbf{D}$, then $\mathbf{C}^* = \mathbf{V}_r \mathbf{\Gamma}$. Because $\mathbf{V}_c^T \mathbf{V}_r = \mathbf{0}$, we have $\mathbf{V}_c^T \mathbf{C}^* = \mathbf{V}_c^T \mathbf{V}_r \mathbf{\Gamma} = \mathbf{0}$. Then, it follows that:

$$\|\mathbf{X}\|_F = \left\| \begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{V}_c^T (\mathbf{X} - \mathbf{C}^*) \end{bmatrix} \right\|_F. \quad (12)$$

Since for any matrixes $\mathbf{M}$ and $\mathbf{N}$ with the same number of columns, it holds that

$$\left\| \begin{bmatrix} \mathbf{M} \\ \mathbf{N} \end{bmatrix} \right\|_F^2 = \|\mathbf{M}\|_F^2 + \|\mathbf{N}\|_F^2. \quad (13)$$

From (12) and (13), we have

$$\|\mathbf{X}\|_F^2 = \|\mathbf{\Gamma}\|_F^2 + \|\mathbf{V}_c^T (\mathbf{X} - \mathbf{C}^*)\|_F^2 \quad (14)$$

which shows that $\|\mathbf{X}\|_F \geq \|\mathbf{\Gamma}\|_F$.

Furthermore, since

$$\|\mathbf{\Gamma}\|_F = \|\mathbf{V}_r \mathbf{\Gamma}\|_F = \|\mathbf{C}^*\|_F \quad (15)$$

we have $\|\mathbf{X}\|_F \geq \|\mathbf{C}^*\|_F$.

Second, we prove that $\mathbf{C}^*$ is the unique solution of (10). Let $\mathbf{X}$ be another minimizer, then, $\mathbf{D} = \mathbf{D}\mathbf{X}$ and $\|\mathbf{X}\|_F = \|\mathbf{C}^*\|_F$. From (14) and (15)

$$\|\mathbf{X}\|_F^2 = \|\mathbf{C}^*\|_F^2 + \|\mathbf{V}_c^T(\mathbf{X} - \mathbf{C}^*)\|_F^2. \quad (16)$$

Since $\|\mathbf{X}\|_F = \|\mathbf{C}^*\|_F$, it must hold that $\|\mathbf{V}_c^T(\mathbf{X} - \mathbf{C}^*)\|_F = 0$, and then $\mathbf{V}_c^T(\mathbf{X} - \mathbf{C}^*) = \mathbf{0}$. Together with $\mathbf{V}_r^T(\mathbf{X} - \mathbf{C}^*) = \mathbf{0}$, this gives $\mathbf{V}^T(\mathbf{X} - \mathbf{C}^*) = \mathbf{0}$. Because $\mathbf{V}$ is an orthogonal matrix, it must hold that $\mathbf{X} = \mathbf{C}^*$. ■

Based on Lemma 1, the following theorem can be used to solve the robust version of PCE (i.e., $\mathbf{E} \neq \mathbf{0}$).

Theorem 1: Let $\mathbf{D} = \mathbf{U}\Sigma\mathbf{V}^T$ be the full SVD of $\mathbf{D} \in \mathbb{R}^{m \times n}$, where the diagonal entries of Σ are in descending order, $\mathbf{U}$ and $\mathbf{V}$ are corresponding left and right singular vectors, respectively. Suppose there exists a clean data set and errors, denoted by $\mathbf{D}_0$ and $\mathbf{E}$, respectively. The optimal $\mathbf{C}$ to (8) is given by $\mathbf{C}^* = \mathbf{V}_k\mathbf{V}_k^T$, where λ is a balanced factor, $\mathbf{V}_k$ consists of the first k right singular vectors of $\mathbf{D}$, $k = \operatorname{argmin}_r r + \lambda \sum_{i>r} \sigma_i^2$, and σ_i denotes the i th diagonal entry of Σ .

Proof: Equation (8) can be rewritten as

$$\min_{\mathbf{D}_0, \mathbf{C}} \frac{1}{2} \|\mathbf{C}\|_F^2 + \frac{\lambda}{2} \|\mathbf{D} - \mathbf{D}_0\|_F^2 \text{ s.t. } \mathbf{D}_0 = \mathbf{D}_0\mathbf{C}. \quad (17)$$

Let $\mathbf{D}_0^* = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^T$ be the skinny SVD of $\mathbf{D}_0$, where r is the rank of $\mathbf{D}_0$. Let $\mathbf{U}_c$ and $\mathbf{V}_c$ be the basis that orthogonal to $\mathbf{U}_r$ and $\mathbf{V}_r$, respectively. Clearly, $\mathbf{I} = \mathbf{V}_r\mathbf{V}_r^T + \mathbf{V}_c\mathbf{V}_c^T$. By Lemma 1, the representation over the clean data $\mathbf{D}_0$ is given by $\mathbf{C}^* = \mathbf{V}_r\mathbf{V}_r^T$. Next, we will bridge $\mathbf{V}_r$ and $\mathbf{V}$.

Using Lagrange method, we have

$$\mathcal{L}(\mathbf{D}_0, \mathbf{C}) = \frac{1}{2} \|\mathbf{C}\|_F^2 + \frac{\lambda}{2} \|\mathbf{D} - \mathbf{D}_0\|_F^2 + \Lambda, \mathbf{D}_0 - \mathbf{D}_0\mathbf{C} > \quad (18)$$

where Λ denotes the Lagrange multiplier and the operator $< \cdot >$ denotes dot product.

Letting $((\partial \mathcal{L}(\mathbf{D}_0, \mathbf{C})) / (\partial \mathbf{D}_0)) = 0$, it gives that

$$\Lambda \mathbf{V}_c \mathbf{V}_c^T = \lambda \mathbf{E}. \quad (19)$$

Letting $((\partial \mathcal{L}(\mathbf{D}_0, \mathbf{C})) / (\partial \mathbf{C})) = 0$, it gives that

$$\mathbf{V}_r \mathbf{V}_r^T = \mathbf{V}_r \mathbf{\Sigma}_r \mathbf{U}_r^T \Lambda. \quad (20)$$

From (20), Λ must be in the form of $\Lambda = \mathbf{U}_r \mathbf{\Sigma}_r^{-1} \mathbf{V}_r^T + \mathbf{U}_c \mathbf{M}$ for some $\mathbf{M}$. Substituting Λ into (19), it gives that

$$\mathbf{U}_c \mathbf{M} \mathbf{V}_c \mathbf{V}_c^T = \lambda \mathbf{E}. \quad (21)$$

Thus, we have $\|\mathbf{E}\|_F^2 = (1/\lambda^2) \|\mathbf{U}_c \mathbf{M} \mathbf{V}_c \mathbf{V}_c^T\|_F^2 = (1/\lambda^2) \|\mathbf{M} \mathbf{V}_c\|_F^2$. Clearly, $\|\mathbf{E}\|_F^2$ is minimized when $\mathbf{M} \mathbf{V}_c$ is a diagonal matrix and can be denoted by $\mathbf{M} \mathbf{V}_c = \mathbf{\Sigma}_c$, i.e., $\mathbf{E} = (1/\lambda) \mathbf{U}_c \mathbf{\Sigma}_c \mathbf{V}_c^T$. Thus, the SVD of $\mathbf{D}$ could be chosen as

$$\mathbf{D} = \mathbf{U}\Sigma\mathbf{V}^T = [\mathbf{U}_r \ \mathbf{U}_c] \begin{bmatrix} \mathbf{\Sigma}_r & \mathbf{0} \\ \mathbf{0} & \frac{1}{\lambda} \mathbf{\Sigma}_c \end{bmatrix} \begin{bmatrix} \mathbf{V}_r^T \\ \mathbf{V}_c^T \end{bmatrix}. \quad (22)$$

Thus, the minimal cost of (17) is given by

$$\begin{aligned} \mathcal{L}_{\min}(\mathbf{D}_0^*, \mathbf{C}^*) &= \frac{1}{2} \|\mathbf{V}_r \mathbf{V}_r^T\|_F^2 + \frac{\lambda}{2} \left\| \frac{1}{\lambda} \mathbf{\Sigma}_c \right\|_F^2 \\ &= \frac{1}{2} r + \frac{\lambda}{2} \sum_{i=r+1}^{\min\{m,n\}} \sigma_i^2 \end{aligned} \quad (23)$$

where σ_i is the i th largest singular value of $\mathbf{D}$. Let k be the optimal r to (23), then we have $k = \operatorname{argmin}_r r + \lambda \sum_{i>r} \sigma_i^2$. ■

Theorem 1 shows that the skinny SVD of $\mathbf{D}$ is automatically separated into two parts, the top and the bottom one correspond to a desired clean data $\mathbf{D}_0$ and the possible corruptions $\mathbf{E}$, respectively. Such a PCA-like result provides a good explanation toward the robustness of our method, i.e., the clean data can be recovered by using the first k leading singular vectors of $\mathbf{D}$. It should be pointed out that the above theoretical results (Lemma 1 and Theorem 1) have been presented in [46] for building the connections between Frobenius norm based representation and nuclear norm based representation in theory. Different from [46], this paper mainly considers how to utilize this result to achieve robust and automatic subspace learning.

Fig. 1 gives an example to show the effectiveness of PCE. We carried out experiment using 700 clean AR facial images [62] as training data that distribute over 100 individuals. Fig. 1(a) shows the coefficient matrix $\mathbf{C}^*$ obtained by PCE. One can find that the matrix is approximately block-diagonal, i.e., $c_{ij} \neq 0$ if and only if the corresponding points $\mathbf{d}_i$ and $\mathbf{d}_j$ belong to the same class. Moreover, we perform SVD over $\mathbf{C}^*$ and show the singular values of $\mathbf{C}^*$ in Fig. 1(b). One can find that only the first 69 singular values are nonzero. In other words, the intrinsic dimension of the entire data set is 69 and the first 69 singular values can preserve 100% information. It should be pointed out that, PCE does not set a parameter to truncate the trivial singular values like PCA and PCA-like methods [19], which incorporates all energy into a small number of dimension.

B. Intrinsic Dimension Estimation and Projection Learning

After obtaining the coefficient matrix $\mathbf{C}^*$, PCE builds a similarity graph and embeds it into an m' -dimensional space by the following NPE [12], that is:

$$\min_{\Theta} \frac{1}{2} \|\Theta^T \mathbf{D} - \Theta^T \mathbf{D} \mathbf{A}\|_F^2, \text{ s.t. } \Theta^T \mathbf{D} \mathbf{D}^T \Theta = \mathbf{I} \quad (24)$$

where $\Theta \in \mathbb{R}^{m \times m'}$ denotes the projection matrix.

One challenging problem arising in dimension reduction is to determine the value of m' , most existing methods experimentally set this parameter, which is very computational inefficiency. To solve this problem, we propose estimating the feature dimension using the rank of the affinity matrix $\mathbf{A}$ and have the following theorem.

Theorem 2: For a given data set $\mathbf{D}$, the feature dimension m' is upper bounded by the rank of $\mathbf{C}^*$, that is

$$m' \leq k. \quad (25)$$

Proof: It is easy to see that (24) has the following equivalent variation:

$$\Theta^* = \operatorname{argmax}_{\Theta} \frac{\Theta^T \mathbf{D} (\mathbf{A} + \mathbf{A}^T - \mathbf{A} \mathbf{A}^T) \mathbf{D}^T \Theta}{\Theta^T \mathbf{D} \mathbf{D}^T \Theta}. \quad (26)$$

We can see that the optimal solution to (26) consists of m' leading eigenvectors of the following generalized Eigen decomposition problem:

$$\mathbf{D} (\mathbf{A} + \mathbf{A}^T - \mathbf{A} \mathbf{A}^T) \mathbf{D}^T \theta = \sigma \mathbf{D} \mathbf{D}^T \theta \quad (27)$$

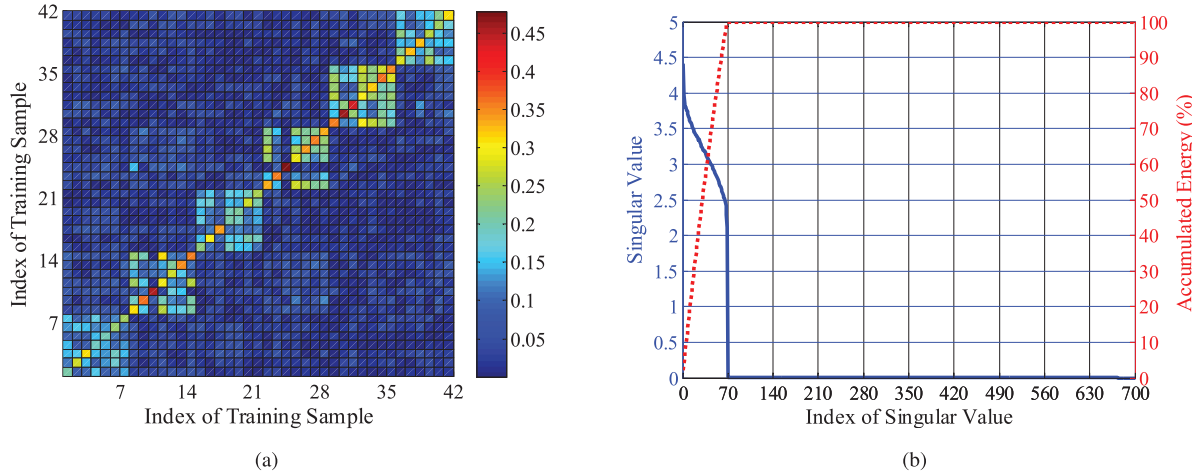


Fig. 1. Illustration using 700 AR facial images. (a) PCE can obtain a block-diagonal affinity matrix, which is benefit to classification. For better illustration, we only show the affinity matrix of the data points belonging to the first seven categories. (b) Intrinsic dimension of the used whole data set is exactly 69, i.e., $m' = k = 69$ for 700 samples. This result is obtained without truncating the trivial singular values like PCA. In Fig. 1(b), the dotted line denotes the accumulated energy of the first k singular value.

Algorithm 1 Automatic Subspace Learning via PCE

Input: A collection of training data points $\mathbf{D} = \{\mathbf{d}_i\}$ sampled from a union of linear subspaces and the balanced parameter $\lambda > 0$.

- 1: Perform the full SVD or skinny SVD on $\mathbf{D}$, i.e. $\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, and get the $\mathbf{C} = \mathbf{V}_k\mathbf{V}_k^T$, where $\mathbf{V}_k$ consists of k column vector of $\mathbf{V}$ corresponding to k largest singular values, where $k = \operatorname{argmin}_r r + \lambda \sum_{i>r} \sigma_i^2(\mathbf{D})$ and $\sigma_i(\mathbf{D})$ is the i -th singular value of $\mathbf{D}$.
- 2: Construct a similarity graph via $\mathbf{A} = \mathbf{C}$.
- 3: Embed $\mathbf{A}$ into a k -dimensional space and get the projection matrix $\mathbf{\Theta} \in \mathbb{R}^{m \times k}$ that consists of the eigenvectors corresponding to the k largest eigenvalues of the following generalized eigenvector problem Eq. (26).

Output: The projection matrix $\mathbf{\Theta}$. For any data point $\mathbf{y} \in \operatorname{span}\{\mathbf{D}\}$, its low-dimensional representation can be obtained by $\mathbf{z} = \mathbf{\Theta}^T \mathbf{y}$.

where σ is the corresponding singular value of the problem.

As $\mathbf{A} = \mathbf{A}^T = \mathbf{A}\mathbf{A}^T$, then (27) can be rewritten

$$\mathbf{D}\mathbf{A}\mathbf{D}^T\mathbf{\Theta} = \sigma\mathbf{D}\mathbf{D}^T\mathbf{\Theta}. \quad (28)$$

From Theorem 1, we have $\operatorname{rank}(\mathbf{D}) > \operatorname{rank}(\mathbf{A}) = k$, where k is calculated according to Theorem 1. Thus, the above generalized Eigen decomposition problem has at most k eigenvalues larger than zeros, i.e., the rank of $\mathbf{\Theta}$ is upperly bounded by k . This gives the result. ■

Algorithm 1 summarizes the procedure of PCE. Note that, it does not require $\mathbf{A}$ to be a symmetric matrix.

C. Computational Complexity Analysis

For a training data set $\mathbf{D} \in \mathbb{R}^{m \times n}$, PCE performs the skinny SVD over $\mathbf{D}$ in $O(m^2n + mn^2 + n^3)$. However, a number of fast SVD methods can speed up this procedure. For example, the complexity can be reduced to $O(mnr)$ by Brand's method [63],

where r is the rank of $\mathbf{D}$. Moreover, PCE estimates the feature dimension k in $O(r \log r)$ and solves a sparse generalized eigenvector problem in $O(mn + mn^2)$ with Lanczos eigensolver. Putting everything together, the time complexity of PCE is $O(mn + mn^2)$ due to $r \ll \min(m, n)$.

IV. EXPERIMENTS AND RESULTS

In this section, we reported the performance of PCE and six state-of-the-art unsupervised feature extraction methods including Eigenfaces [11], LPP [13], [48], NPE [12], L1-graph [15], non-negative matrix factorization (NMF) [64], [65], RPCA [26], NeNMF [66], and robust orthonormal subspace learning (ROSL) [35]. Noticed that, NeNMF is one of the most efficient NMF solvers, which can effectively overcome the slow convergence rate, numerical instability and nonconvergence issue of NMF. All algorithms are implemented in MATLAB. The used data sets and the codes of our algorithm can be downloaded from the website <http://machineilab.org/users/pengxi>.

A. Experimental Setting and Data Sets

We implemented a fast version of L1-graph by using Homotopy algorithm [67] to solve the ℓ_1 -minimization problem. According to [49], Homotopy is one of the most competitive ℓ_1 -optimization algorithms in terms of accuracy, robustness, and convergence speed. For RPCA, we adopted the accelerated proximal gradient method with partial SVD [68] which has achieved a good balance between computation speed and reconstruction error. As mentioned above, RPCA cannot obtain the projection matrix for subspace learning. For fair comparison, we incorporated Eigenfaces with RPCA (denoted by RPCA+PCA) and ROSL (denoted by ROSL+PCA) to obtain the low-dimensional features of the inputs. Unless otherwise specified, we assigned $m' = 300$ for all the tested methods except PCE which automatically determines the value of m' .

TABLE II
USED DATABASES. s AND n_i DENOTE THE NUMBER OF SUBJECT
AND THE NUMBER OF IMAGES FOR EACH GROUP

Databases	s	n_i	Original Size	Cropped Size
AR	100	26	165×120	55×40
ExYaleB	38	58	192×168	54×48
MPIE-S1	249	14	100×82	55×40
MPIE-S2	203	10	100×82	55×40
MPIE-S3	164	10	100×82	55×40
MPIE-S4	176	10	100×82	55×40
COIL100	100	10	128×128	64×64
USPS	10	1100	16×16	-

In our experiments, we evaluated the performance of these subspace learning algorithms with three classifiers, i.e., sparse representation based classification (SRC) [69], [70], support vector machine (SVM) with linear kernel [71], and the NN classifier. For all the evaluated methods, we first identify their optimal parameters using a data partitions and then reported the mean and standard deviation of classification accuracy using ten randomly sampling data partitions.

We used eight image data sets including AR facial database [62], expended Yale database B (ExYaleB) [72], four sessions of multiple PIE (MPIE) [73], COIL100 objects database [74], and the handwritten digital database USPS.¹

The used AR data set contains 2600 samples from 50 male and 50 female subjects, of which 1400 samples are clean images, 600 samples are disguised by sunglasses, and the remaining 600 samples are disguised by scarves. ExYaleB contains 2414 frontal-face images of 38 subjects, and we use the first 58 samples of each subject. MPIE contains the facial images captured in four sessions. In the experiments, all the frontal faces with 14 illuminations² are investigated. For computational efficiency, we downsized all the data sets from the original size to smaller one. Table II provides an overview of the used data sets.

B. Influence of the Parameter

In this section, we investigate the influence of parameters of PCE. Besides the aforementioned subspace clustering methods, we also report the performance of CorrEntropy based sparse representation (CESR) [75] as a baseline. Noticed that, CESR is a not subspace learning method, which performs like SRC to classify each testing sample by finding which subject produces the minimal reconstruction error. By following the experimental setting in [75], we evaluated CESR using the nonnegativity constraint with 0.

1) *Influence of λ* : PCE uses the parameter λ to measure the possible corruptions and estimate the feature dimension m' . To investigate the influence of λ on the classification accuracy and the estimated dimension, we increased the value of λ from 1 to 99 with an interval of 2 by performing experiment on a subset of AR database and a subset of Extended Yale Database B. The used data sets include 1400 clean images over 100 individuals and 2204 samples over 38 subjects. In the experiment, we

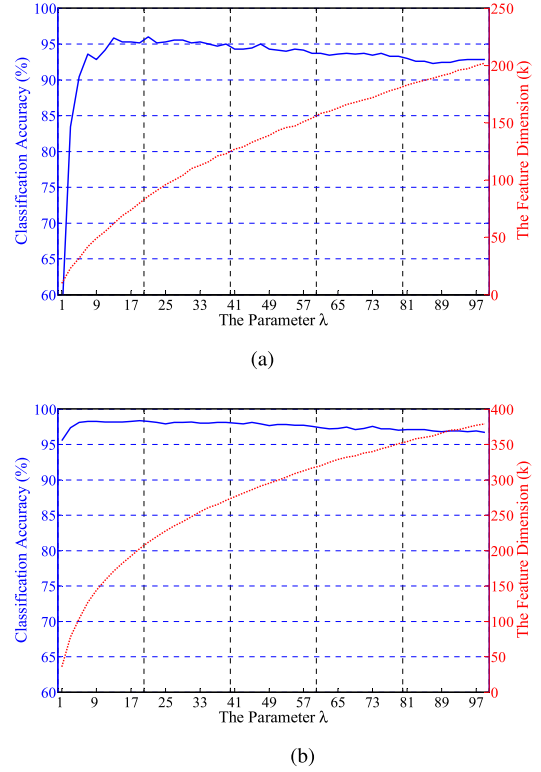


Fig. 2. Influence of the parameter λ , where the NN classifier is used. The solid and dotted lines denote the classification accuracy and the estimated feature dimension m' (i.e., k), respectively. (a) 1400 nondisguised images from the AR database. (b) 2204 images from the ExYaleB database.

randomly divided each data set into two parts with equal size for training and testing.

Fig. 2 shows that a larger λ will lead to a larger m' but does not necessarily bring a higher accuracy since the value of λ does reflect the errors contained into inputs. For example, while λ increases from 13 to 39, the recognition accuracy of PCE on AR almost remains unchanged, which ranges from 93.86% to 95.29%.

2) *PCE With the Fixed m'* : To further show the effectiveness of our dimension determination method, we investigated the performance of PCE by manually specifying $m' = 300$, denoted by PCE2. We carried out the experiments on ExYaleB by choosing 40 samples from each subject as training data and using the rests for testing. Table III reports the result from which we can find the following.

- 1) The automatic version of our method, i.e., PCE, performs competitive to PCE2 which manually set $m' = 300$. This shows that our dimension estimation method can accurately estimate the feature dimension.
- 2) Both PCE and PCE2 outperform the other methods by a considerable performance margin. For example, PCE is 3.68% at least higher than the second best method when the NN classifier is used.
- 3) Although PCE is not the fastest algorithm, it achieves a good balance between recognition rate and computational efficiency. In the experiments, PCE, Eigenfaces, LPP, NPE, and NeNMF are remarkably faster than

¹<http://archive.ics.uci.edu/ml/datasets.html>

²Illuminations: 0, 1, 3, 4, 6, 7, 8, 11, 13, 14, 16, 17, 18, 19.

TABLE III

PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING EXYALEB, WHERE TRAINING DATA AND TESTING DATA CONSIST OF 1520 AND 684 SAMPLES, RESPECTIVELY. PCE, EIGENFACES, AND NMF HAVE ONLY ONE PARAMETER. PCE NEEDS SPECIFYING THE BALANCED PARAMETER λ BUT IT AUTOMATICALLY COMPUTES THE FEATURE DIMENSION. ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION. "PARA." INDICATES THE TUNED PARAMETERS. NOTE THAT, THE SECOND PARAMETER OF PCE DENOTES m' (I.E., k) WHICH IS AUTOMATICALLY CALCULATED VIA THEOREM 1

Classifiers	SRC			SVM			NN		
	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	96.90±0.74	23.50±2.36	5, 118	98.93±0.18	7.44±0.37	50, 329	97.03±0.57	6.96±0.71	5, 118
PCE2	96.92±0.59	28.02±2.84	16.00	98.20±0.43	8.07±0.67	26.00	96.86±0.57	7.89±0.88	19.00
Eigenfaces	95.32±0.80	27.79±0.22	-	95.53±0.85	5.65±0.14	-	82.53±1.70	4.97±0.14	-
LPP	83.87±6.59	17.20±0.71	9.00	87.92±9.12	7.40±0.12	2.00	79.97±1.36	7.18±0.19	3.00
NPE	90.47±15.72	37.80±0.45	50.00	82.50±8.74	27.57±0.24	47.00	93.35±0.53	28.37±0.30	49.00
L1-graph	91.29±0.60	633.95±47.94	1e-2,1e-1	82.08±1.66	870.04±61.01	1e-3,1e-3	89.75±0.70	988.27±74.98	1e-2,1e-3
NMF	87.54±1.15	137.46±6.26	-	91.59±1.09	19.39±0.36	-	72.11±1.44	11.13±0.03	-
NeNMF	87.09±1.11	72.10±6.37	-	76.73±2.14	10.38±0.66	-	47.63±1.03	6.89±0.02	-
RPCA+PCA	95.88±0.56	497.48±32.72	0.30	95.79±1.02	466.17±35.85	0.10	82.57±1.18	466.11±42.70	0.20
ROLS+PCA	95.73±0.77	765.95±15.95	0.23	95.03±0.86	733.46±18.09	0.19	81.33±1.65	732.61±15.58	0.35

TABLE IV

PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING EXYALEB). BESIDES $m' = 300$, ALL METHODS EXCEPT PCE ARE WITH THE TUNED m'

Methods	Fixed m' ($m' = 300$)		The Tuned m'		
	Accuracy	Para.	Accuracy	Para.	m'
PCE	93.51	263	94.63	95	162
Eigenfaces	70.71	-	76.11	-	353
LPP	77.76	3.00	76.73	3	312
NPE	85.18	43.00	87.54	65	405
L1-graph	88.78	1e-2,1e-3	89.18	1e-2,1e-2	532
NMF	62.21	-	70.15	-	214
NeNMF	51.42	-	69.88	-	148
RPCA+PCA	70.86	0.10	76.25	0.1	375
ROLS+PCA	70.46	0.35	89.18	0.39	322
CESR	88.71	1e-3,1e-3	88.85	1e-3,1e-3	336

other baseline methods. Moreover, NeNMF is remarkably faster than NMF while achieving a competitive performance.

3) *Tuning m' for the Baseline Methods:* To show the dominance of the dimension estimation of PCE, we reported the performance of all the baseline methods in two settings, i.e., $m' = 300$ and the optimal m' . The later setting is achieved by finding an optimal m' from 1 to 600 so that the algorithm achieves their highest classification accuracy. We carried out the experiments on ExYaleB by selecting 20 samples from each subject as training data and using the rests for testing. Note that, we only tuned m' for the baseline algorithms and PCE automatically identifies this parameter. Table IV shows that PCE remarkably outperforms the investigated methods in two settings even though all parameters including m' are tuned for achieving the best performance of the baselines.

C. Performance With Increasing Training Data and Feature Dimension

In this section, we examined the performance of PCE with increasing training samples and increasing feature dimension. In the first test, we randomly sampled n_i clean AR images from each subject for training and used the rest for testing. Besides the result of RPCA+PCA, we also reported the performance of RPCA without dimension reduction.

In the second test, we randomly chose a half of images from ExYaleB for training and used the rest for testing. We reported the recognition rate of the NN classifier with the first m' features extracted by all the tested subspace learning methods, where m' increases from 1 to 600 with an interval of 10. From Fig. 3, we can conclude the following.

- 1) PCE performs well even though only a few of training samples are available. Its accuracy is about 90% when $n_i = 5$, whereas the second best method achieves the same accuracy when $n_i = 9$.
- 2) RPCA and RPCA+PCA perform very close, however, RPCA+PCA is more efficient than RPCA.
- 3) Fig. 3(b) shows that PCE consistently outperforms the other methods. This benefits an advantage of PCE, i.e., PCE obtains a more compact representation which can use a few of variables to represent the entire data.

D. Subspace Learning on Clean Images

In this section, we performed the experiments using MPIE and COIL100. For each data set, we split it into two parts with equal size. As did in the above experiments, we set $m' = 300$ for all the tested methods except PCE. Tables V–IX report the results, from which one can find the following.

- 1) With three classifiers, PCE outperforms the other investigated approaches on these five data sets by a considerable performance margin. For example, the recognition rates of PCE with these three classifiers are 6.59%, 5.83%, and 7.90% at least higher than the rates of the second best subspace learning method on MPIE-S1.
- 2) PCE is more stable than other tested methods. Although SRC generally outperforms SVM and NN with the same feature, such superiority is not distinct for PCE. For example, SRC gives an accuracy improvement of 1.02% over NN to PCE on MPIE-S4. However, the corresponding improvement to RPCA+PCA is about 49.50%.
- 3) PCE achieves the best results in all the tests, while using the least time to perform dimension reduction and classification. PCE, Eigenfaces, LPP, NPE, and NeNMF are remarkably efficient than L1-graph, NMF, RPCA+PCA, and ROSL+PCA.

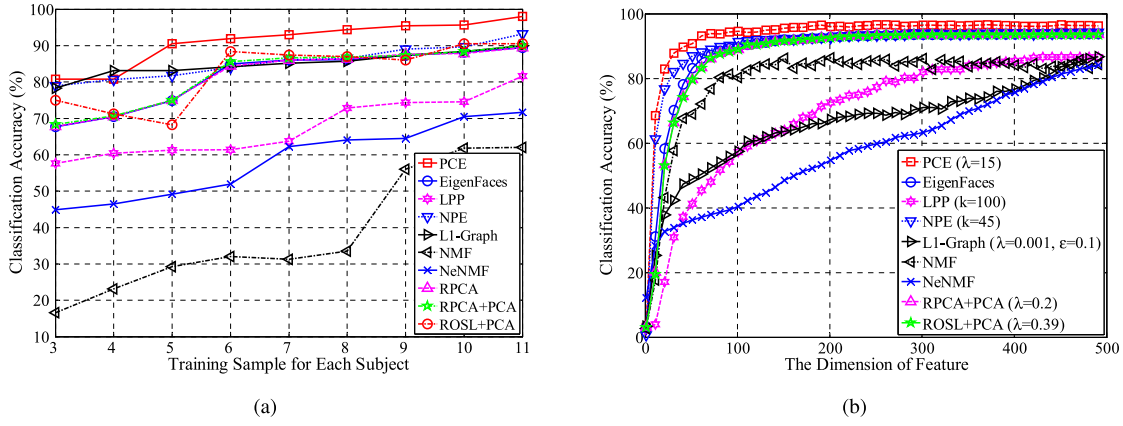


Fig. 3. (a) Performance of the evaluated subspace learning methods with the NN classifier on AR images. (b) Recognition rates of the NN classifier with different subspace learning methods on ExYaleB. Note that, PCE does not automatically determine the feature dimension in the experiment of performance versus increasing feature dimension.

TABLE V
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING THE FIRST SESSION OF MPIE (MPIE-S1).
ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION

Classifiers Algorithms	SRC			SVM			NN		
	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	99.27±0.32	51.96±0.29	75.00	96.56±1.23	14.38±0.52	85.00	97.72±0.55	13.21±0.59	40.00
Eigenfaces	92.64±0.56	90.64±0.73	-	90.73±1.81	12.87±0.20	-	55.03±0.93	6.21±0.22	-
LPP	81.84±0.94	30.58±2.60	10.00	70.16±0.07	7.38±0.54	55.00	71.31±2.39	4.85±0.39	4.00
NPE	80.56±0.41	58.95±0.77	29.00	80.25±0.15	36.38±0.55	43.00	77.71±1.65	36.19±0.38	49.00
L1-graph	80.36±0.17	3856.69±280.16	1e-1, 1e-1	86.79±1.62	5726.08±444.82	1e-6, 1e-5	89.82±1.44	8185.55±503.80	1e-6, 1e-4
NMF	65.18±0.87	520.94±6.27	-	66.42±1.66	121.89±0.58	-	41.78±1.18	11.03±0.00	-
NeNMF	27.42±1.18	277.33±26.87	-	17.57±1.44	36.79±1.83	-	14.66±1.07	10.24±0.04	-
RPCA+PCA	92.68±0.57	1755.27±490.99	0.10	90.51±1.26	1497.13±329.00	0.30	54.95±1.38	1557.33±358.93	0.10
ROLS+PCA	92.39±0.91	658.29±39.47	0.19	89.9±1.71	578.53±38.61	0.35	54.82±1.22	593.07±60.41	0.27

TABLE VI
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING THE SECOND SESSION OF MPIE (MPIE-S2).
ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION

Classifiers Algorithms	SRC			SVM			NN		
	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	93.87±0.82	29.00±0.36	90.00	92.63±0.95	4.97±0.11	40.00	93.18±0.87	4.14±0.14	75.00
Eigenfaces	64.36±2.42	81.71±14.99	-	51.72±2.81	0.50±0.11	-	30.86±1.44	0.36±0.06	-
LPP	59.62±2.33	36.69±7.64	2.00	34.28±2.53	2.73±0.60	2.00	62.64±2.20	2.73±0.84	3.00
NPE	84.65±0.77	33.03±1.51	41.00	64.66±3.03	12.45±0.30	27.00	85.56±0.92	12.24±0.24	49.00
L1-graph	47.67±3.09	874.91±53.69	1e-3, 1e-3	65.41±1.69	657.69±53.51	1e-3, 1e-3	74.15±1.67	703.54±37.97	1e-2, 1e-3
NMF	81.88±1.31	323.93±8.70	-	83.19±1.47	46.72±1.22	-	57.21±1.38	26.01±0.01	-
NeNMF	33.38±1.46	128.33±8.43	-	19.65±1.08	26.79±1.39	-	11.94±0.55	14.58±0.06	-
RPCA+PCA	91.18±1.11	401.62±7.46	0.20	91.18±1.11	401.62±7.46	0.20	67.80±1.93	366.50±8.78	0.10
ROLS+PCA	91.07±0.91	875.79±74.21	0.27	83.02±2.2	633.85±42.38	0.43	43.7±0.98	761.98±67.65	0.23

TABLE VII
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING THE THIRD SESSION OF MPIE (MPIE-S3).
ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION

Classifiers Algorithms	SRC			SVM			NN		
	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	97.79±0.81	13.14±0.19	75.00	95.37±1.82	2.74±0.08	65.00	94.04±0.84	2.29±0.04	65.00
Eigenfaces	88.04±0.70	29.51±0.36	-	80.99±2.28	2.01±0.05	-	37.96±1.18	0.94±0.05	-
LPP	78.73±2.04	28.61±4.62	40.00	60.44±2.49	1.61±0.25	3.00	65.96±2.49	1.03±0.13	75.00
NPE	77.83±3.14	25.79±1.02	46.00	72.29±0.99	7.56±0.07	7.00	79.18±2.38	7.06±0.09	48.00
L1-graph	70.40±0.22	1315.37±192.65	1e-1, 1e-5	79.28±2.54	1309.27±193.38	1e-3, 1e-3	89.40±2.80	1539.26±226.57	1e-3, 1e-3
NMF	60.94±0.80	90.64±0.91	-	51.34±1.68	40.04±0.37	-	39.89±1.04	4.28±0.01	-
NeNMF	39.90±1.19	61.24±3.21	-	26.66±1.79	20.30±0.41	-	11.93±0.95	3.11±0.01	-
RPCA+PCA	88.49±2.17	630.08±88.89	0.10	81.02±2.52	491.36±26.75	0.30	37.85±0.83	481.87±25.01	0.30
ROLS+PCA	87.13±1.53	327.14±30.97	0.15	78.29±3.09	291.57±1.64	0.47	37.23±1.24	297.75±24.91	0.35

E. Subspace Learning on Corrupted Facial Images

In this section, we investigated the robustness of PCE against two corruptions using ExYaleB and the NN classifier.

The corruptions include the white Gaussian noise (additive noise) and the random pixel corruption (nonadditive noise) [69].

TABLE VIII
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING THE FOURTH SESSION OF MPIE (MPIE-S4).
ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION

Classifiers Algorithms	SRC			SVM			NN		
	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	98.36±0.41	14.07±0.31	100.00	90.55±1.02	3.04±0.12	45.00	97.34±0.78	2.73±0.09	80.00
Eigenfaces	92.05±1.37	32.43±0.32	-	82.18±3.88	2.34±0.05	-	43.74±1.17	1.12±0.05	-
LPP	64.67±2.52	27.38±1.57	3.00	61.47±1.12	1.94±0.20	2.00	73.69±2.68	1.11±0.17	2.00
NPE	84.74±1.50	30.45±1.28	46.00	63.80±1.56	9.87±0.49	49.00	87.30±1.10	8.54±0.36	45.00
L1-graph	70.45±0.31	1928.24±212.21	1e-3,1e-3	84.67±2.46	1825.09±197.62	1e-3,1e-3	93.56±1.13	1767.57±156.61	1e-3,1e-3
NMF	69.41±1.73	98.91±1.37	-	53.48±2.07	47.26±0.44	-	25.47±1.40	4.85±0.00	-
NeNMF	40.61±1.21	58.45±3.76	-	23.78±1.80	20.87±1.17	-	14.83±0.61	3.36±0.02	-
RPCA+PCA	93.16±1.17	682.27±39.20	0.30	84.45±3.02	535.31±19.08	0.10	43.66±0.63	514.51±20.82	0.10
ROLS+PCA	91.8±1.01	200.83±25.47	0.07	82.61±2.12	265.63±7.03	0.23	43.01±1.58	264.21±6.46	0.27

TABLE IX
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING COIL100.
ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION

Classifiers Algorithms	SRC			SVM			NN		
	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	59.60±1.94	12.25±0.32	15.00	53.00±1.22	1.36±0.01	45.00	57.40±1.83	1.15±0.03	5.00
Eigenfaces	57.40±1.67	12.97±0.25	-	44.40±2.21	1.04±0.06	-	54.76±1.14	0.67±0.06	-
LPP	45.86±1.51	13.22±0.54	60.00	30.20±3.08	0.80±0.11	2.00	41.10±2.15	0.63±0.02	90.00
NPE	47.72±2.25	15.30±0.28	43.00	32.78±2.90	5.33±0.08	36.00	44.88±2.12	6.81±0.03	49.00
L1-graph	45.16±1.83	960.80±123.43	1e-2,1e-4	39.42±2.81	801.73±147.83	1e-3,1e-3	38.06±1.96	664.92±93.75	1e-1,1e-5
NMF	51.42±2.17	76.05±1.21	-	41.74±2.05	32.81±0.18	-	56.82±1.46	6.47±0.00	-
NeNMF	57.48±2.13	39.21±3.18	-	35.96±3.73	25.64±0.52	-	59.02±1.55	10.99±0.01	-
RPCA+PCA	58.04±0.90	244.92±50.17	0.30	45.52±2.70	229.54±51.06	0.20	56.48±1.32	227.27±52.66	0.10
ROLS+PCA	58.11±1.67	447.05±27.01	0.03	44.74±1.69	747.66±98.89	0.19	57.10±1.68	379.81±18.42	0.03

TABLE X
PERFORMANCE OF DIFFERENT SUBSPACE LEARNING ALGORITHMS WITH THE NN CLASSIFIER USING THE CORRUPTED EXYALEB. ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION. RPC IS THE SHORT FOR RANDOM PIXEL CORRUPTION. THE NUMBER IN THE PARENTHESES DENOTES THE LEVEL OF CORRUPTION

Corruptions Algorithms	Gaussian (10%)		Gaussian (30%)		RPC (10%)		RPC (30%)	
	Accuracy	Para.	Accuracy	Para.	Accuracy	Para.	Accuracy	Para.
PCE	95.05±0.63	10.00	93.18±0.87	5.00	90.12±0.98	5.00	83.48±1.04	10.00
Eigenfaces	41.69±2.01	-	30.86±1.44	-	30.35±2.05	-	25.37±1.56	-
LPP	76.94±0.75	2.00	62.64±2.20	3.00	55.86±1.27	2.00	42.76±1.53	2.00
NPE	91.54±0.76	49.00	85.56±0.92	49.00	80.66±0.86	49.00	60.25±1.64	43.00
L1-graph	87.36±0.81	1e-3,1e-4	74.15±1.67	1e-2,1e-3	71.63±0.90	1e-3,1e-4	55.02±2.07	1e-4,1e-4
NMF	67.42±1.41	-	57.21±1.38	-	60.57±1.88	-	46.13±1.41	-
NeNMF	44.75±1.28	-	42.48±0.68	-	43.27±1.01	-	25.44±1.21	-
RPCA+PCA	76.26±1.12	0.20	67.80±1.93	0.10	64.56±0.67	0.10	52.12±1.34	0.10
ROSL+PCA	76.52±0.83	0.27	66.50±1.49	0.27	75.94±1.44	0.07	65.76±0.98	0.07

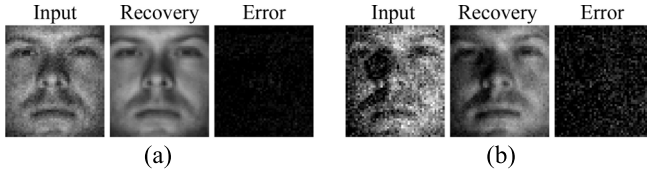


Fig. 4. Some results achieved by PCE over the corrupted ExYaleB data set which is corrupted by the Gaussian noise. The recovery and the error are identified by PCE according to Theorem 1. (a) Corruption ratio: 10%. (b) Corruption ratio: 30%.

In our experiments, we use a half of images (29 images per subject) to corrupt using these two noises. Specifically, we added white Gaussian noise into the sampled data $\mathbf{d}$ via $\tilde{\mathbf{d}} = \mathbf{d} + \rho \mathbf{n}$, where $\tilde{\mathbf{d}} \in [0 \ 255]$, ρ is the corruption ratio, and $\mathbf{n}$ is the noise following the standard normal distribution. For random pixel corruption, we replaced the value of a percentage of pixels randomly selected from the image with the values

following a uniform distribution over $[0, p_{\max}]$, where $p_{\max}$ is the largest pixel value of $\mathbf{d}$. After adding the noises into the images, we randomly divide the data into training and testing sets. In other words, both training data and testing data probably contains corruptions. Fig. 4 illustrates some results achieved by our method. We can see that PCE successfully identifies the noises from the corrupted samples and recovers the clean data. Table X reports the comparison from which we can see the following.

- 1) PCE is more robust than the other tested approaches. When 10% pixels are randomly corrupted, the accuracy of PCE is at least 9.46% higher than that of the other methods.
- 2) With the increase of level of noise, the dominance of PCE is further strengthened. For example, the improvement in accuracy of PCE increases from 9.46% to 23.23% when ρ increases to 30%.

TABLE XI
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING THE AR IMAGES DISGUISED BY SUNGLASSES.
ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION

Classifiers Algorithms	SRC			SVM			NN		
	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	83.57±1.16	8.95±0.96	50.00	87.70±1.62	0.90±0.11	95.00	66.92±1.75	0.69±0.10	50.00
Eigenfaces	69.32±2.58	36.18±3.97	-	63.93±2.84	1.42±0.34	-	30.58±1.27	0.71±0.19	-
LPP	49.48±1.79	30.80±6.29	2.00	43.07±1.80	1.34±0.39	2.00	33.70±1.70	0.85±0.21	90.00
NPE	62.75±2.16	19.44±0.62	47.00	58.23±2.75	4.40±0.03	49.00	54.33±2.37	4.09±0.03	49.00
L1-graph	49.65±1.42	4340.22±453.64	1e-3, 1e-4	48.53±2.06	5381.96±467.89	1e-4, 1e-4	49.28±2.68	5189.73±411.98	1e-4, 1e-4
NMF	47.17±2.18	109.33±2.22	-	40.55±2.20	23.67±0.92	-	24.58±1.88	5.82±0.01	-
NeNMF	32.58±1.76	52.20±1.81	-	21.98±2.10	14.71±0.08	-	18.52±1.46	3.49±0.00	-
RPCA+PCA	71.40±2.67	241.45±4.39	0.10	66.40±2.62	193.67±7.76	0.10	32.27±1.46	195.21±8.01	0.20
ROLS+PCA	68.57±1.37	557.54±92.71	0.15	60.38±2.53	441.79±95.61	0.23	31.03±1.05	432.02±95.28	0.07

TABLE XII
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING THE AR IMAGES DISGUISED BY SCARVES.
ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION

Classifiers Algorithms	SRC			SVM			NN		
	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	83.88±1.38	8.73±0.90	65.00	87.80±1.57	0.90±0.10	65.00	68.58±1.96	0.71±0.11	55.00
Eigenfaces	72.87±1.99	45.48±5.18	-	64.77±2.96	1.62±0.40	-	36.42±1.69	0.78±0.19	-
LPP	51.73±2.77	44.20±8.25	95.00	44.88±1.93	1.60±0.57	2.00	37.37±2.19	1.12±0.30	85.00
NPE	78.00±2.27	19.84±0.51	47.00	49.17±3.33	4.30±0.04	47.00	56.83±1.83	4.16±0.04	49.00
L1-graph	52.00±1.42	4340.22±573.64	1e-4, 1e-4	48.53±2.06	3899.81±487.89	1e-4, 1e-4	49.28±2.68	4189.73±431.98	1e-4, 1e-4
NMF	47.87±2.64	108.46±2.98	-	43.05±2.39	24.34±0.81	-	31.35±2.04	8.01±0.01	-
NeNMF	37.57±2.41	55.36±4.31	-	26.17±2.30	15.14±0.34	-	26.52±1.43	7.62±0.01	-
RPCA+PCA	72.07±2.30	1227.08±519.27	0.10	63.70±3.74	1044.46±462.33	0.20	36.93±0.90	965.76±385.19	0.10
ROLS+PCA	71.42±1.46	1313.65±501.05	0.31	62.67±3.27	1336.00±549.66	0.43	35.58±2.14	1245.63±406.74	0.19

F. Subspace Learning on Disguised Facial Images

Besides the above tests on the robustness to corruptions, we also investigated the robustness to real disguises. Tables XI and XII reports results on two subsets of AR database. The first subset contains 600 clean images and 600 images disguised with sunglasses (occlusion rate is about 20%), and the second one includes 600 clean images and 600 images disguised by scarves (occlusion rate is about 40%). Like the above experiment, both training data and testing data will contains the disguised images. From the results, one can conclude the following.

- 1) PCE significantly outperforms the other tested methods. When the images are disguised by sunglasses, the recognition rates of PCE with SRC, SVM, and NN are 5.88%, 23.03%, and 11.75% higher than the best baseline method. With respect to the images with scarves, the corresponding improvements are 12.17%, 21.30%, and 17.64%.
- 2) PCE is one of the most computationally efficient methods. When SRC is used, PCE is 2.27 times faster than NPE and 497.16 times faster than L1-graph on the faces with sunglasses. When the faces are disguised by scarves, the corresponding speedup are 2.17 and 484.94 times, respectively.

G. Comparisons With Some Dimension Estimation Techniques

In this section, we compare PCE with three dimension estimators, i.e., maximum likelihood estimation [76], minimum neighbor distance Estimators (MiNDs) [77], and DANCo [78]. MiND has two variants which are denoted as MiND-ML and

TABLE XIII
PERFORMANCE OF DIFFERENT DIMENSION ESTIMATORS WITH THE NN CLASSIFIER, WHERE m' DENOTES THE ESTIMATED FEATURE DIMENSION AND ONLY THE TIME COST (SECOND) FOR DIMENSION ESTIMATION IS TAKEN INTO CONSIDERATION

Methods	Accuracy	Time Cost	m'	Para.
PCE	90.31	0.57	109	30
MLE+PCA	68.29	3.34	11.6	10
MiND-ML+PCA	67.14	3.95	11	10
MiND-KL+PCA	72.71	2791.19	16	22
DANCo+PCA	71.71	28804.01	15	22

MiND-KL. All these estimators need specifying the size of neighborhood of which the optimal value is found from the range of [10 30] with an interval of 2. Since these estimators cannot be used for dimension reduction, we report the performance of these estimators with PCA, i.e., we first estimate the feature dimension with an estimator and then extract features using PCA with the estimated dimension. We carry out experiments with the NN classifier on a subset of the AR data set of which both the training and testing set include 700 nondisguised facial images. Table XIII shows that our approach outperforms the baseline estimators by a considerable performance margin in terms of classification accuracy and time cost.

H. Scalability Evaluation

In this section, we investigate the scalability performance of PCE by using the whole USPS data set, where λ of PCE is fixed as 0.05. In the experiments, we randomly split the whole data set into two partitions for training and testing, where the

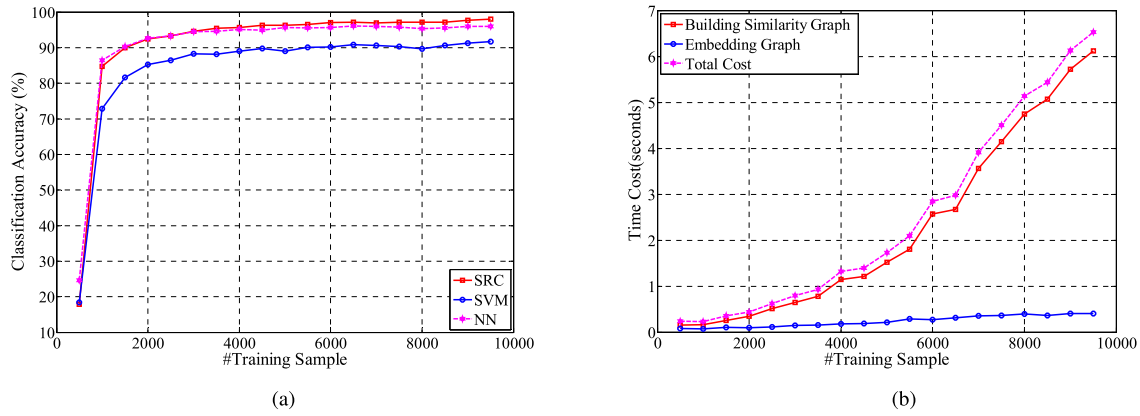


Fig. 5. Scalability performance of PCE on the whole USPS data set, where the number training samples increase from 500 to 9500 with an interval of 500. (a) Recognition rate of PCE with three classifiers. (b) Time costs for different steps of PCE, where total cost is the cost for building similarity graph and embedding graph.

number of training samples increases from 500 to 9500 with an interval of 500 and thus 19 partitions are obtained. Fig. 5 reports the classification accuracy and the time cost taken by PCE. From the results, we could see that the recognition rate of PCE almost remains unchanged when 1500 samples are available for training. Considering different classifiers, SRC slightly performs better than NN, and both of them remarkably outperform SVM. PCE is computational efficient, it only take about seven seconds to handle 9500 samples. Moreover, PCE could be further speeded up by adopting large scale SVD methods. However, this has been out of scope for this paper.

V. CONCLUSION

In this paper, we have proposed a novel unsupervised subspace learning method, called PCE. Unlike existing subspace learning methods, PCE can automatically determine the optimal dimension of feature space and obtain the low-dimensional representation of a given data set. Experimental results on several popular image databases have shown that our PCE achieves a good performance with respect to additive noise, nonadditive noise, and partial disguised images.

This paper would be further extended or improved from the following aspects. First, this paper currently only considers one category of image recognition, i.e., image identification. In the future, PCE can be extended to handle the other category of image recognition, i.e., face verification which aims to determine whether a given pair of facial images is from the same subject or not. Second, PCE is a unsupervised method which does not adopt the label information. If such information is available, one can develop the supervised or semi-supervised version of PCE under the framework of graph embedding. Third, PCE can be extended to handle outliers by enforcing $\ell_{2,1}$ -norm or Laplacian noises by enforcing ℓ_1 -norm over the errors term in our objective function.

ACKNOWLEDGMENT

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

REFERENCES

- [1] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugen.*, vol. 7, no. 2, pp. 179–188, 1936.
- [2] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. 18th Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2004, pp. 513–520.
- [3] 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.*, vol. 2, San Diego, CA, USA, Jun. 2005, pp. 846–853.
- [4] J. Lu, X. Zhou, Y.-P. Tan, Y. Shang, and J. Zhou, "Neighborhood repulsed metric learning for kinship verification," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 36, no. 2, pp. 331–345, Feb. 2014.
- [5] H. Wang, X. Lu, Z. Hu, and W. Zheng, "Fisher discriminant analysis with L1-norm," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 828–842, Jun. 2014.
- [6] 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, Aug. 2016.
- [7] C. Xu, D. Tao, and C. Xu, "Large-margin multi-view information bottleneck," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1559–1572, Aug. 2014.
- [8] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proc. 11th Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–7.
- [9] S. Yan and H. Wang, "Semi-supervised learning by sparse representation," in *Proc. 9th SIAM Int. Conf. Data Mining*, Sparks, NV, USA, Apr. 2009, pp. 792–801.
- [10] C. Xu, D. Tao, C. Xu, and Y. Rui, "Large-margin weakly supervised dimensionality reduction," in *Proc. 31th Int. Conf. Mach. Learn.*, Beijing, China, Jun. 2014, pp. 865–873.
- [11] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.
- [12] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proc. 10th IEEE Conf. Comput. Vis.*, Beijing, China, Oct. 2005, pp. 1208–1213.
- [13] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.
- [14] L. S. Qiao, S. C. Chen, and X. Y. Tan, "Sparsity preserving projections with applications to face recognition," *Pattern Recognit.*, vol. 43, no. 1, pp. 331–341, 2010.
- [15] 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.
- [16] C. Xu, D. Tao, and C. Xu, "Multi-view intact space learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2531–2544, Dec. 2015.
- [17] S. C. Yan *et al.*, "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.

- [18] M. Polito and P. Perona, "Grouping and dimensionality reduction by locally linear embedding," in *Proc. 14th Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2001, pp. 1255–1262.
- [19] S. Yan, J. Liu, X. Tang, and T. S. Huang, "A parameter-free framework for general supervised subspace learning," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 1, pp. 69–76, Mar. 2007.
- [20] B. Kégl, "Intrinsic dimension estimation using packing numbers," in *Proc. 15th Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada 2002, pp. 681–688.
- [21] F. Camastra and A. Vinciarelli, "Estimating the intrinsic dimension of data with a fractal-based method," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 24, no. 10, pp. 1404–1407, Oct. 2002.
- [22] D. Mo and S. H. Huang, "Fractal-based intrinsic dimension estimation and its application in dimensionality reduction," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 1, pp. 59–71, Jan. 2012.
- [23] P. Mordohai and G. Medioni, "Dimensionality estimation, manifold learning and function approximation using tensor voting," *J. Mach. Learn. Res.*, vol. 11, pp. 411–450, Mar. 2010.
- [24] K. M. Carter, R. Raich, and A. O. Hero, III, "On local intrinsic dimension estimation and its applications," *IEEE Trans. Signal Process.*, vol. 58, no. 2, pp. 650–663, Feb. 2010.
- [25] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, pp. 119–155, Dec. 2003.
- [26] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, 2011, Art. no. 11.
- [27] R. He, B.-G. Hu, W.-S. Zheng, and X.-W. Kong, "Robust principal component analysis based on maximum coreentropy criterion," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1485–1494, Jun. 2011.
- [28] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.
- [29] Q. Zhao, D. Meng, Z. Xu, W. Zuo, and L. Zhang, "Robust principal component analysis with complex noise," in *Proc. 31th Int. Conf. Mach. Learn.*, Beijing, China, Jun. 2014, pp. 55–63.
- [30] F. Nie, J. Yuan, and H. Huang, "Optimal mean robust principal component analysis," in *Proc. 31th Int. Conf. Mach. Learn.*, Beijing, China, Jun. 2014, pp. 1062–1070.
- [31] X. Liu, Y. Mu, D. Zhang, B. Lang, and X. Li, "Large-scale unsupervised hashing with shared structure learning," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1811–1822, Sep. 2015.
- [32] D. Hsu, S. M. Kakade, and T. Zhang, "Robust matrix decomposition with sparse corruptions," *IEEE Trans. Inf. Theory*, vol. 57, no. 11, pp. 7221–7234, Nov. 2011.
- [33] B.-K. Bao, G. Liu, R. Hong, S. Yan, and C. Xu, "General subspace learning with corrupted training data via graph embedding," *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4380–4393, Nov. 2013.
- [34] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Subspace learning from image gradient orientations," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2454–2466, Dec. 2012.
- [35] X. Shu, F. Porikli, and N. Ahuja, "Robust orthonormal subspace learning: Efficient recovery of corrupted low-rank matrices," in *Proc. 27th IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 3874–3881.
- [36] T. Liu and D. Tao, "On the robustness and generalization of Cauchy regression," in *Proc. IEEE Int. Conf. Info. Sci. Technol.*, Shenzhen, China, Apr. 2014, pp. 100–105.
- [37] T. Liu and D. Tao, "On the performance of manhattan nonnegative matrix factorization," *IEEE Trans. Neural. Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2015.2458986.
- [38] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Trans Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 447–461, Mar. 2016.
- [39] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2011, pp. 1801–1807.
- [40] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognit. Lett.*, vol. 43, pp. 47–61, Jul. 2014.
- [41] X. Peng, Z. Yi, and H. Tang, "Robust subspace clustering via thresholding ridge regression," in *Proc. 29th AAAI Conf. Artif. Intell.*, Austin, TX, USA, Jan. 2015, pp. 3827–3833.
- [42] S. Jones and L. Shao, "Unsupervised spectral dual assignment clustering of human actions in context," in *Proc. 27th IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 604–611.
- [43] S. Xiao, M. Tan, and D. Xu, "Weighted block-sparse low rank representation for face clustering in videos," in *Proc. 13th Eur. Conf. Comput. Vis.*, Zürich, Switzerland, 2014, pp. 123–138.
- [44] Z. Yu *et al.*, "Generalized transitive distance with minimum spanning random forest," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Buenos Aires, Argentina, Jul. 2015, pp. 2205–2211.
- [45] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2016.2536752.
- [46] X. Peng, C. Lu, Z. Yi, and H. Tang, "Connections between nuclear norm and Frobenius norm based representation," *CoRR*, vol. abs/1502.07423, 2015. [Online]. Available: <http://arxiv.org/abs/1502.07423>
- [47] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [48] X. He and P. Niyogi, "Locality preserving projections," in *Proc. 17th Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2004, pp. 153–160.
- [49] A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Fast L1-minimization algorithms and an application in robust face recognition: A review," Dept. EECS, Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2010-13, Feb. 2010.
- [50] Q. Liu and J. Wang, "L1-minimization algorithms for sparse signal reconstruction based on a projection neural network," *IEEE Trans. Neural. Netw. Learn. Syst.*, vol. 27, no. 3, pp. 698–707, Mar. 2016.
- [51] Z. Zhang, S. Yan, and M. Zhao, "Pairwise sparsity preserving embedding for unsupervised subspace learning and classification," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4640–4651, Dec. 2013.
- [52] J. Tang, L. Shao, X. Li, and K. Lu, "A local structural descriptor for image matching via normalized graph Laplacian embedding," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 410–420, Feb. 2016.
- [53] J. Lu, G. Wang, W. Deng, and K. Jia, "Reconstruction-based metric learning for unconstrained face verification," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 79–89, Jan. 2015.
- [54] D. Tao, L. Jin, Z. Yang, and X. Li, "Rank preserving sparse learning for Kinect based scene classification," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1406–1417, Oct. 2013.
- [55] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 793–804, Jun. 2014.
- [56] S.-B. Chen, C. H. Q. Ding, and B. Luo, "Similarity learning of manifold data," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1744–1756, Sep. 2015.
- [57] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, 2010.
- [58] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. 13th IEEE Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 471–478.
- [59] H. Zhang, Z. Yi, and X. Peng, "FLRR: Fast low-rank representation using Frobenius-norm," *Electron. Lett.*, vol. 50, no. 13, pp. 936–938, 2014.
- [60] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.
- [61] G. Liu *et al.*, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
- [62] A. Martinez and R. Benavente, "The AR face database," Centre de Visió per Computador, Universitat Autònoma de Barcelona, Barcelona, Spain, Tech. Rep. 24, 1998.
- [63] M. Brand, "Fast low-rank modifications of the thin singular value decomposition," *Linear Algebra Appl.*, vol. 415, no. 1, pp. 20–30, 2006.
- [64] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Dec. 2004.
- [65] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Manifold regularized discriminative nonnegative matrix factorization with fast gradient descent," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 2030–2048, Jul. 2011.
- [66] N. Guan, D. Tao, Z. Luo, and B. Yuan, "NeNMF: An optimal gradient method for nonnegative matrix factorization," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2882–2898, Jun. 2012.
- [67] M. R. Osborne, B. Presnell, and B. A. Turlach, "A new approach to variable selection in least squares problems," *SIAM J. Numer. Anal.*, vol. 20, no. 3, pp. 389–403, 2000.
- [68] Z. Lin *et al.*, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," UIUC, Tech. Rep. UILU-ENG-09-2214, Champaign, IL, USA, Aug. 2009.

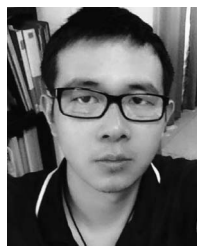
- [69] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
- [70] S. Gao, K. Jia, L. Zhuang, and Y. Ma, "Neither global nor local: Regularized patch-based representation for single sample per person face recognition," *Int. J. Comput. Vis.*, vol. 111, no. 3, pp. 365–383, 2014.
- [71] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.
- [72] 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.
- [73] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image Vis. Comput.*, vol. 28, no. 5, pp. 807–813, 2010.
- [74] S. K. Nayar, S. A. Nene, and H. Murase, "Columbia object image library (COIL 100)," Dept. Comput. Sci., Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-006-96, 1996.
- [75] R. He, W.-S. Zheng, and B.-G. Hu, "Maximum correntropy criterion for robust face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1561–1576, Aug. 2011.
- [76] E. Levina and P. J. Bickel, "Maximum likelihood estimation of intrinsic dimension," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2004, pp. 777–784.
- [77] G. Lombardi, A. Rozza, C. Ceruti, E. Casiraghi, and P. Campadelli, "Minimum neighbor distance estimators of intrinsic dimension," in *Proc. Eur. Conf. Mach. Learn.*, vol. 6912. Athens, Greece, 2011, pp. 374–389.
- [78] C. Ceruti *et al.*, "DANCo: An intrinsic dimensionality estimator exploiting angle and norm concentration," *Pattern Recognit.*, vol. 47, no. 8, pp. 2569–2581, 2014.



Jiwen Lu (M'11–SM'15) received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, and the Ph.D. degree in electrical engineering from the Nanyang Technological University, Singapore, in 2003, 2006, and 2011, respectively.

He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China. From 2011 to 2015, he was a Research Scientist with the Advanced Digital Sciences Center, Singapore. His current research interests include computer vision, pattern recognition, and machine learning. He has authored/co-authored over 130 scientific papers in the above areas, where over 50 papers are published in the IEEE TRANSACTIONS and top-tier computer vision conferences.

Dr. Lu was a recipient of the First-Prize National Scholarship and the National Outstanding Student Award from the Ministry of Education of China in 2002 and 2003, the Best Student Paper Award from Pattern Recognition and Machine Intelligence Association of Singapore in 2012, the Top 10% Best Paper Award from the IEEE International Workshop on Multimedia Signal Processing in 2014, and the National 1000 Young Talents Plan Program in 2015, respectively. He serves/has served as an Associate Editor of *Pattern Recognition Letters*, *Neurocomputing*, the *Journal of Signal Processing Systems*, the *IEEE Access* and the *IEEE Biometrics Council Newsletters*, a Guest Editor of *Pattern Recognition*, *Computer Vision and Image Understanding*, *Image and Vision Computing*, and *Neurocomputing*, and an elected member of the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society. He is/was the Area Chair of over ten international conference such as IEEE Winter Conference on Applications of Computer Vision (WACV) and has given tutorials at several international conferences including IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15).



Xi Peng received the B.Eng. degree in electronic engineering and the M.Eng. degree in computer science from the Chongqing University of Posts and Telecommunications, Chongqing, China, and the Ph.D. degree from Sichuan University, Chengdu, China, respectively.

He is currently a Research Scientist with the Institute for Infocomm, Research Agency for Science, Technology and Research (A*STAR), Singapore. His current research interests include machine intelligence and computer vision.

Dr. Peng was a recipient of the Excellent Graduate Student of Sichuan Province, the China National Graduate Scholarship, the CSC-IBM Scholarship for Outstanding Chinese Students, and the Excellent Student Paper of IEEE CHENGDU Section. He has served as a Guest Editor of *Image and Vision Computing*, a PC Member/Reviewer for ten international conferences such as *AAAI Conference on Artificial Intelligence*, and a Reviewer for over ten international journals.



Zhang Yi (SM'10–F'15) received the Ph.D. degree in mathematics from the Institute of Mathematics, Chinese Academy of Science, Beijing, China, in 1994.

He is currently a Professor with the Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu, China. He is the co-author of three books: *Convergence Analysis of Recurrent Neural Networks* (Kluwer Academic Publishers, 2004), *Neural Networks: Computational Models and Applications* (Springer, 2007), and *Subspace Learning of Neural Networks* (CRC Press, 2010). His current research interests include neural networks and big data.

Prof. Yi was an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS from 2009 to 2012, and since 2014, he has been an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS.



Rui Yan (M'11) received the bachelor's and master's degrees from the Department of Mathematics, Sichuan University, Chengdu, China, in 1998 and 2001, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, in 2006.

She is a Professor with the College of Computer Science, Sichuan University. Her current research interests include intelligent robots, neural computation, and nonlinear control.