An Evolutionary Orthogonal Component Analysis Method for Incremental Dimensionality Reduction

Tianyue Zhang, Furao Shen, *Member, IEEE* , Tao Zhu, and Jian Zhao, *Senior Member, IEEE*

***Abstract*— In order to quickly discover the low-dimensional representation of high-dimensional noisy data in online environments, we transform the linear dimensionality reduction problem into the problem of learning the bases of linear feature subspaces. Based on that, we propose a fast and robust dimensionality reduction framework for incremental subspace learning named evolutionary orthogonal component analysis (EOCA). By setting adaptive thresholds to automatically determine the target dimensionality, the proposed method extracts the orthogonal subspace bases of data incrementally to realize dimensionality reduction and avoids complex computations. Besides, EOCA can merge two learned subspaces that are represented by their orthonormal bases to a new one to eliminate the outlier effects, and the new subspace is proved to be unique. Extensive experiments and analysis demonstrate that EOCA is fast and achieves competitive results, especially for noisy data.**

***Index Terms*— Dimensionality reduction, incremental learning, orthogonal component (OC), subspace learning.**

I. Introduction

THE contemporary resourceful Internet and multiple data acquisition techniques provide researchers with a vast amount of inherently high-dimensional data, which enables us to complete more complex and difficult tasks than before. However, one big problem in manipulating those data is "the curse of dimensionality," which is induced by high-dimensional spaces. Processing a small amount of data in high dimensionality can encounter many problems in data analysis [1], including invalid distance norm and difficulty in optimization. Thus, we need to reduce the data dimensionality and discover the intrinsic data space bases in order to facilitate subsequent data research and visualization.

Dimensionality reduction is a powerful and indispensable tool for data analysis, which represents high-dimensional data in low dimensions. Linear dimensionality reduction methods aim to achieve that goal through a linear transformation. Based on the assumption that the data of interest lie in an embedded linear subspace, the linear dimensionality reduction problem can be transformed to the problem of linear subspace learning. Linear dimensionality reduction methods have been developed over a century due to their simple geometric interpretations and typically attractive computational properties [2]. The earliest and most widely used unsupervised method, i.e., principal component analysis (PCA) [3], learns an orthogonal linear subspace basis by maximizing data discrimination or minimizing reconstruction error. Supervised methods, such as linear discriminant analysis (LDA) [4], aim to find linear subspaces to realize the maximal data discrimination with labeled information. Nonnegative matrix factorization (NMF) [5] decomposes the original data matrix into the basis matrix and the coefficient matrix so that the original data can be reconstructed with minimal error. To make the bases more interpretable, NMF restricts all the factored matrices to be nonnegative.

To employ nonlinear mappings, some nonlinear dimensionality reduction methods are presented. Isomap methods [6], [7] try to keep the global geometric features of the input data by exploiting geodesic paths. Locally linear embedding method [8] reduces the dimensionality using the neighborhood-preserving mapping. More complex and deep models, such as autoencoder [9], [10], are employed to learn the embedded subspace to reduce the dimension of data, which is suitable for bigger data sets. Besides, dimensionality reduction methods have been widely employed in real-world applications, such as hyperspectral image classification [11], [12], robot control [13], speaker identification [14], and sentiment analysis [15]. Theoretical analysis is given [16] to discuss the difference between the nuclear norm and the Frobenius norm used for objective functions, which provides us with suggestions for selecting the appropriate norm for different applications.

Recently, incremental learning models have become more important in online scenarios [17], in which stream-like data are fed to the system from sensors. Such data are usually of high dimensions. Moreover, the data analysis models are required to adapt to data of incremental samples and incremental inherent (target) dimensionality with environmental noise and to realize real-time response.

Most of the current incremental methods are derived from existing linear dimension reduction methods that develop

Manuscript received July 12, 2019; revised January 28, 2020, May 31, 2020, and September 2, 2020; accepted September 24, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61876076, in part by Jiangsu NSF under Grant BK20171344, and in part by the Dengfeng Scholars Program of Nanjing University under Grant B1512002. *(Corresponding authors: Furao Shen; Jian Zhao.)*

Tianyue Zhang, Furao Shen, and Tao Zhu are with the National Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China (e-mail: njucszty@gmail.coml; frshen@nju.edu.cn; tao144@gmail.com).

Jian Zhao is with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China (e-mail: jianzhao@nju.edu.cn).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

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

Digital Object Identifier 10.1109/TNNLS.2020.3027852

updating strategies to adjust the basis matrices with new samples. Incremental PCA and previous works [18] only preserve the required principal components and their coefficients. They are updated by newly arrived samples. The widely used incremental PCA method proposed in [19] is able to merge and split eigenspaces in the learning process. The candid covariance-free incremental PCA (CCIPCA) algorithm reduces the computation complexity by avoiding computing covariance matrices [20]. Generalized PCA (GPCA) achieves incremental dimension reduction via the QR decomposition [21]. More recently, incremental bidirectional principal component analysis (BDPCA) transfers the eigenvalue decomposition problem of scatters to the singular value decomposition (SVD) of the corresponding unfolded matrices [22]. Those methods have been successfully employed in applications, such as pattern recognition and image analysis [23]. Another supervised subspace learning method, i.e., LDA, also develops its online version. Sequential incremental LDA proposed in [24], which is among the first proposed incremental LDA methods, solves the problem of updating for scatter matrices. Almost at the same time, an incremental dimension reduction algorithm via QR decomposition (IDR/QR) [25] is proposed, which applies QR decomposition to obtain the optimal projection matrix in the subspace. To handle the inverse of the within-class scatter matrix with the SVD technique, generalized SVD LDA (GSVD-ILDA) [26] is proposed and applied to face recognition data sets. The incremental LDA algorithm employs the concept of sufficient spanning set in approximation [27]. Besides, incremental DCV methods [28], [29] are also presented to deal with the dimensionality reduction problem. Incremental orthogonal projective nonnegative matrix factorization (IOPNMF) [30] provides a multiplicative updating rule to minimize its objective function. A more thorough overview of the series of incremental subspace learning methods is provided in [31]. However, those methods need to predetermine the target dimensionality and run slowly, which are not practical in dynamic environments.

Some robust incremental linear dimensionality reduction methods, such as in [32], [33], are proposed to deal with outliers. The more recent work, i.e., the incremental orthogonal component (OC) analysis (IOCA) [34], utilizes an adaptive threshold policy to achieve high-speed incremental OC learning, as well as automatic target dimension estimation and updating. The IOCA method gradually obtains numerically OCs through component learning and spends considerably less time on computation compared with other methods. Unfortunately, unknown corrupted values frequently occur in the real-time data acquisition process. Under those circumstances, IOCA cannot adjust the eigenbases that have been already learned, which makes it vulnerable to the outliers, especially in the situation where the outliers emerge at the beginning of the learning process.

Motivated by the abovementioned work, we intend to provide a new online incremental high-speed OC learning model to solve the problems of automatic target estimation and the initial outliers' problem simultaneously in linear dimensionality reduction. We propose a new learning framework, named evolutionary OC analysis (EOCA). EOCA combines the properties of two main component updating strategies in the existing literature on subspace learning: 1) updating the bases of the feature subspace once a new data are fed in and 2) generating auxiliary subspaces based on the input data, and then, employing auxiliary subspaces to update the bases of the feature subspace. Meanwhile, the orthogonality of the learned OCs is guaranteed in both the extracting and the updating process. Intuitively, updating the feature space with auxiliary subspaces can be converted to the problem of rotating the feature space toward the auxiliary subspace.

The main contributions of this work are as follows.

1) We propose a novel online orthonormal subspace updating framework that can be directly employed on arbitrary subspaces that are represented by their orthonormal basis without storing the eigenvalues, thereby the effect of outliers on the OC learning model is decreased.
2) An incremental dimensionality reduction method EOCA based on the online subspace updating algorithm combined with the OC extraction method is proposed; thus, the target estimation and the initial outliers problem are tackled simultaneously.
3) Detailed analysis of the uniqueness of the linear subspace guarantees the reliability of EOCA. The computational complexity analysis and comparison illustrate that the method is a fast learning method. Thorough experiments on the synthetic data set and public benchmark data sets demonstrate the effectiveness of EOCA.

The rest of this article is organized as follows. Section II gives the problem setting and outlines two works that are closely related to our work. Section III introduces our methods. Section IV gives the analysis and findings on the learning process of EOCA, including the linear subspace uniqueness and the computational complexity. Section V is dedicated to experimental results on both synthetic and real-world data sets. Conclusions are drawn in Section VI.

II. Preliminaries

In this section, we briefly present the problem setting and notation of incremental component analysis. To facilitate the understanding of our proposed method, we introduce two related previous work: OC analysis (OCA) [35] and IOCA [34].

A. Problem Setting and Notation

The incremental component analysis can be described as follows. The initial data set is empty, i.e., $\mathbf{X} = \emptyset$. The d-dimensional input samples arrive sequentially, denoted as $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t \in \mathbb{R}^d$. For convenience, we suppose there are N samples, and thus, the input data matrix is represented as $\mathbf{X} \in \mathbb{R}^{d\times N}$. The task is to learn k basis ($k < d$) $\boldsymbol{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_k\} \in \mathbb{R}^{d\times k}$ ($\boldsymbol{b}_i \in \mathbb{R}^d$, $\forall i = 1, \ldots, k$) to represent the feature (low-dimensional) space $\mathcal{S}$ of the data stream. Each sample $\boldsymbol{x}_t$ is represented as a linear combination of the k basis vectors, i.e., $\boldsymbol{x}_t = \boldsymbol{B}\boldsymbol{y}_t$, where $\boldsymbol{y}_t$ is the k-dimensional coefficient vector. Thus, we reduce the original d dimensions of input data to k dimensions.

B. Orthogonal Component Analysis

OCA is a batch dimensionality reduction method and aims to compute $\boldsymbol{X} \approx \boldsymbol{BY}$ for low-dimensional representations. Moreover, OCA learns the OCs with low computational cost and high stability and automatically determines the number of components, i.e., the target dimension. As a fast method, modified Gram–Schmidt (GS) (MGS) orthogonalization process is employed to yield the basis set $\boldsymbol{B}$ of subspace $\mathcal{S}$ in some episodes. The main algorithm can be described as follows:

$$\boldsymbol{r}_i^{(0)} = \boldsymbol{x}_i, i = \arg\max_{l=1,\ldots,N} \left\|\boldsymbol{r}_l^{(t)}\right\|_2^2 \tag{1}$$

$$\boldsymbol{b}_{t+1} = \frac{\boldsymbol{r}_i^{(t)}}{\left\|\boldsymbol{r}_i^{(t)}\right\|_2}, \boldsymbol{r}_l^{(t+1)} = \boldsymbol{r}_l^{(t)} - \sum_{j=1}^{t} \boldsymbol{b}_j \boldsymbol{b}_j^{\top} \boldsymbol{r}_l^{(t)}, \quad l = 1, \ldots N \tag{2}$$

where t represents the learned dimension of subspace $\mathcal{S}$, and we initialize $t = 0$. Compared with traditional GS method, MGS is numerically more stable. To guarantee the orthogonality of the basis, $\boldsymbol{x}_i$ is chosen with the maximal $\|\boldsymbol{r}_i^{(t+1)}\|_2$ as the $(t+1)$th principal component when t bases are already learned and preserved in the basis matrix $\boldsymbol{B}$. An adaptive threshold, which compares $\boldsymbol{r}_{t+1}$ and the amount of basis in the subspace matrix, determines whether the algorithm stops, that is,

$$\frac{\|\boldsymbol{r}_{t+1}\|_2}{\|\boldsymbol{r}_1\|_2} \geq f\left(\frac{t}{d}\right). \tag{3}$$

If (3) is satisfied, $\boldsymbol{b}_{t+1}$ is added to the basis matrix $\boldsymbol{B}$; otherwise, the learning process stops. $f(\omega)$ is required to be a strictly monotonic increasing function, and $0 \leq f(\omega) \leq 1$ when $0 \leq \omega \leq 1$.

C. Incremental Orthogonal Component Analysis

IOCA is an incremental dimensionality reduction method based on OCA, which inherits the advantages of low computational cost and automatic target dimension estimation. Since incoming samples may increase the target dimension in online environments, the latter feature is more important in online learning than in batch learning. IOCA is able to automatically extract the desired OCs for subspace generating. The number of extracted components, i.e., the dimension of the feature subspace, can be adaptively determined during the learning process.

The principle of orthonormal component extraction is that the new orthonormal component is extracted only when the linear independence between the currently input data vector $\boldsymbol{x}_t \in \mathbb{R}^d$ and the learned k-dimensional subspace $\mathcal{S}$ is greater than the threshold T. According to the linear dependence theorem, it can be measured by $\|\boldsymbol{r}_t\|_2$, i.e., the projection distance from $\boldsymbol{x}_t$ to $\mathcal{S}$. To achieve adaptive intrinsic dimension estimation, the threshold T should be balanced between the expansion and the maintenance of subspace $\mathcal{S}$. When the dimension of $\mathcal{S}$ is small, the algorithm tends to learn more information from the input data through component extraction. As the dimension of $\mathcal{S}$ increases, preventing blind expansion of $\mathcal{S}$ becomes more important, i.e., the difficulty of accepting new components should increase as the dimension of $\mathcal{S}$ increases. As a result, the adaptive threshold derived from (3) is modified without the knowledge of the whole data set, which is described as

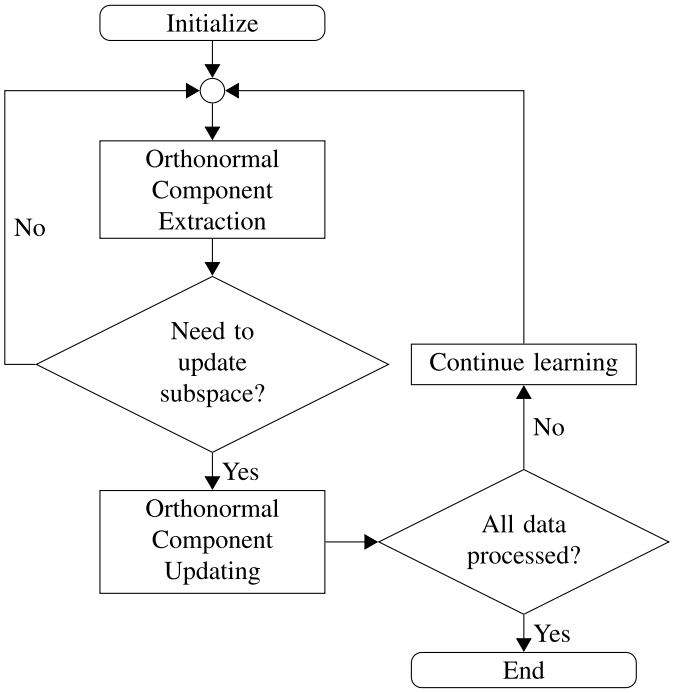


Fig. 1. Framework of the proposed EOCA algorithm. Two phase methods, i.e., orthonormal component extraction and orthonormal component updating, are combined to learn new bases.

$$\frac{\|\boldsymbol{r}_t\|_2}{L_{\max}^{(t)}} \geq f\left(\frac{k}{d}\right) \tag{4}$$

where $L_{\max}^{(t)} = \max\{\|\boldsymbol{x}_1\|_2, \ldots, \|\boldsymbol{x}_t\|_2\}$. Theoretical analyses of the process are given in [34].

We now detail the component extraction procedure. In the beginning, subspace $\mathcal{S}$ is initialized as a 0-D space. When the tth data $\boldsymbol{x}_t$ are fed as the input, assuming $\mathcal{S} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\}$ and $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ to be orthonormal, a candidate orthonormal component $\boldsymbol{b}_{k+1}$ is extracted from $\boldsymbol{x}_t$ based on the GS process. Instead of MGS, GS is more suitable for online environment, which is described as

$$\boldsymbol{r}_1 = \boldsymbol{x}_1, \boldsymbol{b}_1 = \frac{\boldsymbol{r}_1}{\|\boldsymbol{r}_1\|_2} \tag{5}$$

$$\boldsymbol{r}_t = \boldsymbol{x}_t - \sum_{j=1}^{t-1} \boldsymbol{b}_j \boldsymbol{b}_j^{\top} \boldsymbol{x}_t, \quad \boldsymbol{b}_t = \frac{\boldsymbol{r}_t}{\|\boldsymbol{r}_t\|_2}, \quad t = 2, \ldots, N. \tag{6}$$

If the threshold described in (4) is satisfied, $\boldsymbol{b}_{k+1}$ is added into the basis set of $\mathcal{S}$, and its dimension increases; otherwise, $\boldsymbol{b}_{k+1}$ is discarded. Then, the algorithm continues to process the next input data until all the data are processed. With the help of the adaptive threshold policy, $\mathcal{S}$ achieves a steady state during the learning process. Moreover, the orthogonality of bases is guaranteed.

D. Orthogonality

The reasons why orthogonality is desired for the bases of the embedded subspaces in the previous works and our EOCA method are summarized as follows. First, some works have proved that orthogonal bases are more appropriate for dimension reduction, such as in [36] and [37]. Many methods [37], [38] are improved with an orthogonal basis inspired by the idea. Second, when the norm of the bases is normalized to the same value, i.e. 1, the orthonormal bases are more linearly independent, and the condition number of the basis matrix is more appropriate, which leads to more stable computational results on computers. Finally, it is more convenient to compute reduced-dimensional vectors through orthonormal bases. If the bases are not orthogonal, we need to compute the matrix inversion for reduced vectors, while we only need the transpose orthogonal basis matrices. In a word, orthogonal brings simple computation and better performance. The experiments in Section V exhibit a similar conclusion.

III. Evolutionary Orthogonal Component Analysis

A. Overview of the Approach

We use Fig. 1 to illustrate the whole framework of EOCA. Different from OCA and IOCA, the input data are first extracted in an auxiliary subspace, and then, the auxiliary subspace is used to update the feature subspace. In order to distinguish the two spaces, we denote the auxiliary subspace in the learning process as $\mathcal{S}'$ and its basis set as $\boldsymbol{B}_2 = [\boldsymbol{b}_{2,1}, \ldots, \boldsymbol{b}_{2,k_2}] \in \mathbb{R}^{d\times k_2}$. The feature space is denoted as $\mathcal{S}$, and its basis set is $\boldsymbol{B}_1 = [\boldsymbol{b}_{1,1}, \ldots, \boldsymbol{b}_{1,k_1}] \in \mathbb{R}^{d\times k_1}$.

The whole method is composed of two procedures, i.e., orthonormal component extraction and orthonormal component updating. The orthonormal component extraction procedure learns new bases and adds them to a basis set $\boldsymbol{B}_2$ of the auxiliary subspace. The orthonormal component updating procedure uses $\boldsymbol{B}_2$ to update the basis set $\boldsymbol{B}_1$ of the feature space. In the following sections, we detail the design of the two methods.

B. Orthonormal Component Extraction

We employ a similar method as IOCA [34] for extracting orthonormal bases, which has been introduced in Section II-C. Now, we discuss some details of the method when employing it in EOCA.

We require $f(\omega)$ to be a strictly monotonic increasing function, and we have $0 \le f(\omega) \le 1$ when $0 \le \omega \le 1$. How to determine the function is a practical problem. We compare the choice of three common functions in Section V-A.

The learning process is similar to IOCA, except it is conducted on the auxiliary subspace $\mathcal{S}'$, i.e., $\boldsymbol{B}_2 = [\boldsymbol{b}_{2,1}, \ldots, \boldsymbol{b}_{2,k_2}] \in \mathbb{R}^{d\times k_2}$. Thus, k in (4) is replaced by k_2. Note that the procedure avoids solving the matrix eigenproblem or the matrix inversion problem. Thus, its time complexity is low. However, eigenvectors and eigenvalues provide information about the important directions of data, and omitting this computation may lose this information. The solution is to capture the information in the following orthonormal component updating part. When the eigenvalue is big, the adjustment of the corresponding basis is small, which means that the most important directions are preserved during the adjustment procedure. Thus, we only need eigenvalue decomposition when the two subspaces need to be combined, which saves time for our method.

By extracting orthonormal components from continuous data streams, $\mathcal{S}'$ is obtained in an incremental way, and its dimension is automatically determined. However, once an orthonormal component has been extracted, it will never be adjusted. When $\mathcal{S}'$ achieves a steady state, the threshold is strict enough to make sure that the new data rarely influence the current $\mathcal{S}'$. To make full use of the input data and eliminate the effect brought by outliers, we introduce an orthonormal component updating method: if $\mathcal{S}'$ stays unchanged during a certain period of time, we deem it achieves a steady state and update the existing feature subspace $\mathcal{S}$ using it; then, we restart the process of learning $\mathcal{S}'$ through component extraction.

C. Orthonormal Component Updating

From the above sections, we know that $\mathcal{S}$ is the existing k_1-dimensional feature subspace. When new data come, a k_2-dimensional subspace $\mathcal{S}'$ is extracted, and then, we can merge $\mathcal{S}'$ and $\mathcal{S}$ to obtain the updated feature subspace $\mathcal{S}^{(\text{new})}$ whose dimension is k. The column vectors of $\boldsymbol{B}_1 = [\boldsymbol{b}_{1,1}, \ldots, \boldsymbol{b}_{1,k_1}] \in \mathbb{R}^{d\times k_1}$ and $\boldsymbol{B}_2 = [\boldsymbol{b}_{2,1}, \ldots, \boldsymbol{b}_{2,k_2}] \in \mathbb{R}^{d\times k_2}$ are orthonormal bases of $\mathcal{S}$ and $\mathcal{S}'$, respectively. The obtained $\mathcal{S}^{(\text{new})}$ is also represented by its orthonormal basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$.

Intuitively, we hope that the problem of merging two subspaces can be decomposed into a series of subproblems: each time we select a basis vector pair $\boldsymbol{b}_{1,i}$ and $\boldsymbol{b}_{2,i}$ from $\mathcal{S}$ and $\mathcal{S}'$, respectively, the basis vector $\boldsymbol{b}_i$ of $\mathcal{S}^{(\text{new})}$ can be calculated exclusively based on $\boldsymbol{b}_{1,i}$ and $\boldsymbol{b}_{2,i}$ as

$$\boldsymbol{b}_i = \eta \boldsymbol{b}_{1,i} + (1-\eta)\boldsymbol{b}_{2,i} \tag{7}$$

where η is the learning rate. Therefore, basis updating can be conducted by directly calculating each vector pair.

However, there is trouble with the abovementioned strategy: for a nonzero finite-dimensional linear subspace, the representation for its bases is not unique. As a result, given $\mathcal{S}$ and $\mathcal{S}'$, if we do not determine the unique one-to-one relationship between their bases before vector updating, the obtained $\mathcal{S}^{(\text{new})}$ may be nonunique. For example, both $\mathcal{S} = \text{span}\{\boldsymbol{b}_{1,1}\}$ and $\mathcal{S}' = \text{span}\{\boldsymbol{b}_{2,1}\}$ are 1-dimensional subspaces, where $\|\boldsymbol{b}_{1,1}\|_2 = \|\boldsymbol{b}_{2,1}\|_2 = 1$. $\mathcal{S}'$ can also be represented by $\text{span}\{-\boldsymbol{b}_{2,1}\}$. Thus, when $\boldsymbol{b}_{1,1}$ and $\boldsymbol{b}_{2,1}$ are linearly independent, $\text{span}\{(1-\eta)\boldsymbol{b}_{1,1} + \eta\boldsymbol{b}_{2,1}\}$ and $\text{span}\{(1-\eta)\boldsymbol{b}_{1,1} - \eta\boldsymbol{b}_{2,1}\}$ are not the same linear subspace. Therefore, basis alignment should be performed to guarantee the one-to-one relationship before basis adjustment, as illustrated in Fig. 2. The two stages, i.e., basis alignment and basis adjustment, will be introduced in this section.

1) Basis Alignment: We employ the principal angles [39] between subspaces $\mathcal{S}$ and $\mathcal{S}'$ to determine the unique relationships between them. Suppose that $0 \le \theta_1 \le \cdots \le \theta_{\min(k_1,k_2)} \le (\pi/2)$ are the sequence of principal angles between $\mathcal{S}$ and $\mathcal{S}'$,

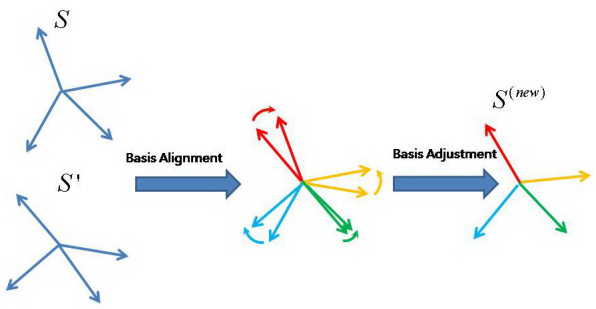


Fig. 2. Basis updating process. The subspaces $\mathcal{S}$ (represented by basis matrix $\boldsymbol{B}_1$) and $\mathcal{S}'$ (represented by basis matrix $\boldsymbol{B}_2$) are aligned by their principal angles, and then, $\mathcal{S}$ and $\mathcal{S}'$ are merged to obtain the updated subspace $\mathcal{S}^{(\text{new})}$.

and the first θ_1 is defined as

$$\cos\theta_1 = \max_{\boldsymbol{w}_{1,1}\in\mathcal{S},\boldsymbol{w}_{2,1}\in\mathcal{S}'} \boldsymbol{w}_{1,1}^\top \boldsymbol{w}_{2,1} \tag{8}$$

where $\|\boldsymbol{w}_{1,1}\|_2 = 1$ and $\|\boldsymbol{w}_{2,1}\|_2 = 1$. Then, the other principal angles are defined recursively via

$$\cos\theta_i = \max_{\boldsymbol{w}_{1,i}\in\mathcal{S},\ \boldsymbol{w}_{2,i}\in\mathcal{S}'} \boldsymbol{w}_{1,i}^\top \boldsymbol{w}_{2,i} \tag{9}$$

$$\text{s.t. } \boldsymbol{w}_{1,i}^\top \boldsymbol{w}_{1,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

$$\boldsymbol{w}_{2,i}^\top \boldsymbol{w}_{2,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j. \end{cases} \tag{10}$$

Based on the principal angles, the bases of $\mathcal{S}$ and $\mathcal{S}'$ can be aligned. Let $k_{min} = \min(k_1, k_2)$, for $i = 1, \ldots, k_{min}$, and we would like to find $\boldsymbol{u}_{1,i}$ and $\boldsymbol{u}_{2,i}$ to make the angle between $\boldsymbol{w}_{1,i} = \boldsymbol{B}_1\boldsymbol{u}_{1,i}$ and $\boldsymbol{w}_{2,i} = \boldsymbol{B}_2\boldsymbol{u}_{2,i}$ to be θ_i, which is the ith principal angle between $\mathcal{S}$ and $\mathcal{S}'$. In other words, we hope to have $\boldsymbol{w}_{1,i}^\top \boldsymbol{w}_{2,i} = \cos\theta_i$, where $\|\boldsymbol{w}_{1,i}\|_2 = \|\boldsymbol{w}_{2,i}\|_2 = 1$.

According to the definition of principal angles, the first θ_1 satisfies

$$\cos\theta_1 = \max_{\boldsymbol{w}_{1,1}\in\mathcal{S},\boldsymbol{w}_{2,1}\in\mathcal{S}'} \boldsymbol{w}_{1,1}^\top \boldsymbol{w}_{2,1}$$

$$= \max_{\boldsymbol{B}_1\boldsymbol{u}_{1,1}\in\mathcal{S},\boldsymbol{B}_2\boldsymbol{u}_{2,1}\in\mathcal{S}'} \boldsymbol{u}_{1,1}^\top \boldsymbol{B}_1^\top \boldsymbol{B}_2 \boldsymbol{u}_{2,1} \tag{11}$$

$$\text{s.t. } \|\boldsymbol{u}_{1,i}\|_2 = \|\boldsymbol{u}_{2,i}\|_2 = 1 \tag{12}$$

where $\boldsymbol{u}_{1,1}$ and $\boldsymbol{u}_{2,1}$ are the vectors that we should obtain. Based on the Lagrange multiplier method, the problem of calculating $\boldsymbol{w}_{1,i}^\top \boldsymbol{w}_{2,i} = \cos\theta_i$ can be transformed into the optimization problem

$$\max \boldsymbol{u}_{1,1}^\top \boldsymbol{B}_1^\top \boldsymbol{B}_2 \boldsymbol{u}_{2,1} + \lambda_{1,1}\left(1 - \boldsymbol{u}_{1,1}^\top \boldsymbol{u}_{1,1}\right) + \lambda_{2,1}\left(1 - \boldsymbol{u}_{2,1}^\top \boldsymbol{u}_{2,1}\right). \tag{13}$$

The above problem can be solved by finding the largest singular value σ_1 of the $k_1 \times k_2$ size matrix $\boldsymbol{B}_1^\top \boldsymbol{B}_2$, while $\boldsymbol{u}_{1,1}$ and $\boldsymbol{v}_{2,1}$ are the left- and right-singular vectors for σ_1.

Similarly, we obtain the other basis pairs one by one. If $\boldsymbol{u}_{1,i}$ and $\boldsymbol{u}_{2,i}$ are the left- and right-singular vectors corresponding to the ith largest singular value of matrix $\boldsymbol{B}_1^\top \boldsymbol{B}_2$, the angle between $\boldsymbol{w}_{1,i} = \boldsymbol{B}_1\boldsymbol{u}_{1,i}$ and $\boldsymbol{w}_{2,i} = \boldsymbol{B}_2\boldsymbol{u}_{2,i}$ forms the ith principal angle θ_i.

Therefore, basis alignment can be achieved through SVD. We obtain

$$\boldsymbol{B}_1^\top \boldsymbol{B}_2 = \boldsymbol{U}_1 \boldsymbol{\Sigma} \boldsymbol{U}_2^\top \tag{14}$$

where $\boldsymbol{U}_1 = [\boldsymbol{u}_{1,1}, \ldots, \boldsymbol{u}_{1,k_1}]$ and $\boldsymbol{U}_2 = [\boldsymbol{u}_{2,1}, \ldots, \boldsymbol{u}_{2,k_2}]$ are $k_1 \times k_1$ and $k_2 \times k_2$ size orthogonal matrices, respectively. $\boldsymbol{\Sigma}$ is a diagonal $k_1 \times k_2$ matrix with nonnegative real numbers on the diagonal. Then, we calculate $\boldsymbol{W}_1 = \boldsymbol{B}_1\boldsymbol{U}_1$ and $\boldsymbol{W}_2 = \boldsymbol{B}_2\boldsymbol{U}_2$ and obtain $\boldsymbol{W}_1 = [\boldsymbol{w}_{1,i}, \ldots, \boldsymbol{w}_{1,k_1}] \in \mathbb{R}^{d\times k_1}$ and $\boldsymbol{W}_2 = [\boldsymbol{w}_{2,i}, \ldots, \boldsymbol{w}_{2,k_2}] \in \mathbb{R}^{d\times k_2}$.

Note that the abovementioned SVD-based alignment operation is similar to the canonical correlation analysis (CCA) [39], but the main difference between CCA and the proposed basis alignment algorithm is that the former is performed on the data vectors in two data sets and the latter is performed on the orthonormal bases of two linear subspaces.

The columns of $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ satisfy

$$\boldsymbol{w}_{1,i}^\top \boldsymbol{w}_{1,j} = \boldsymbol{u}_{1,i}^\top (\boldsymbol{B}_1^\top \boldsymbol{B}_1)\boldsymbol{u}_{1,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{15}$$

$$\boldsymbol{w}_{2,i}^\top \boldsymbol{w}_{2,j} = \boldsymbol{u}_{2,i}^\top (\boldsymbol{B}_2^\top \boldsymbol{B}_2)\boldsymbol{u}_{2,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{16}$$

$$\boldsymbol{w}_{1,i}^\top \boldsymbol{w}_{2,j} = \boldsymbol{u}_{1,i}^\top (\boldsymbol{B}_1^\top \boldsymbol{B}_2)\boldsymbol{u}_{2,j} = \begin{cases} \sigma_i, & i = j \\ 0, & i \neq j. \end{cases} \tag{17}$$

According to the abovementioned properties, we reach the conclusion that vectors in basis sets $\{\boldsymbol{w}_{1,1}, \ldots, \boldsymbol{w}_{1,k_1}\}$ and $\{\boldsymbol{w}_{2,1}, \ldots, \boldsymbol{w}_{2,k_2}\}$ are paired: when $i \neq j$, $\boldsymbol{w}_{1,i}$, and $\boldsymbol{w}_{2,j}$ are orthogonal to each other. As a result, each basis vector pair can be processed separately in the basis updating.

2) Basis Adjustment: Based on the aligned bases, we perform the basis adjustment.

When $\sigma_i = \boldsymbol{w}_{1,i}^\top \boldsymbol{w}_{2,i} > 0$, i.e., the angle between $\boldsymbol{w}_{1,i}$ and $\boldsymbol{w}_{2,i}$ is acute, we can calculate $\mathcal{S}^{(\text{new})}$'s basis vector $\boldsymbol{w}_i$ by

$$\boldsymbol{w}_i = \eta \boldsymbol{w}_{1,i} + (1 - \eta)\boldsymbol{w}_{2,i}. \tag{18}$$

$\mathcal{S}$ and $\mathcal{S}'$ are generated by processing N_1 and N_2 data, respectively. According to the assumption that each datum carries the same significance, we employ

$$\eta = \frac{N_1}{N_1 + N_2} \tag{19}$$

as the default setting for the learning rate η. Then, we have

$$\boldsymbol{w}_i^\top \boldsymbol{w}_j = (\eta \boldsymbol{w}_{1,i} + (1 - \eta)\boldsymbol{w}_{2,i})^\top (\eta \boldsymbol{w}_{1,j} + (1 - \eta)\boldsymbol{w}_{2,j})$$

$$= \begin{cases} \eta^2 + 2\eta(1 - \eta)\sigma_i + (1 - \eta)^2, & i = j \\ 0, & i \neq j. \end{cases} \tag{20}$$

In other words, when $i \neq j$, $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ are also orthogonal to each other. As η^2 and $(1 - \eta)^2$ cannot be 0 at the same time, we have $\eta(1 - \eta)\sigma_i \geq 0$ and $\|\boldsymbol{w}_i\|_2 > 0$. Therefore, we can obtain $\boldsymbol{b}_i = (\boldsymbol{w}_i / \|\boldsymbol{w}_i\|_2)$, and let it be the orthonormal basis vector of $\mathcal{S}^{(\text{new})}$.

When $\sigma_i = \boldsymbol{w}_{1,i}^\top \boldsymbol{w}_{2,i} = 0$, i.e., the angle between $\boldsymbol{w}_{1,i}$ and $\boldsymbol{w}_{2,i}$ is a right angle, we draw the conclusion that the basis computed by (18) is not unique through the analysis of uniqueness in Section IV-A. In this situation, we can either discard or keep both of them. To retain more information, we decide to add both $\boldsymbol{w}_{1,i}$ and $\boldsymbol{w}_{2,i}$ into the basis set of

Algorithm 1 Algorithm of Orthonormal Component Updating

Input: Basis matrix $\boldsymbol{B}_1 \in \mathbb{R}^{d\times k_1}$ and $\boldsymbol{B}_2 \in \mathbb{R}^{d\times k_2}$, the number of data N_1 and N_2 to generate $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ respectively

$[\boldsymbol{U}_1, \boldsymbol{\Sigma}, \boldsymbol{U}_2] = \text{svd}(\boldsymbol{B}_1^\top \boldsymbol{B}_2)$;
$k = 0$, $N = N_1 + N_2$;
for $i = 1, \ldots, k_{min}$ **do**
  if $[\boldsymbol{\Sigma}]_{i,i} > 0$ **then**
    $\boldsymbol{w}_{k+1} = \frac{N_1}{N}\boldsymbol{B}_1\boldsymbol{U}_1(:,i) + \frac{N_2}{N}\boldsymbol{B}_2\boldsymbol{U}_2(:,i)$;
    $\boldsymbol{B}(:,k+1) = \frac{\boldsymbol{w}_{k+1}}{\|\boldsymbol{w}_{k+1}\|_2}$;
    $k = k + 1$;
  else
    $\boldsymbol{B}(:,k+1) = \boldsymbol{B}_1\boldsymbol{U}_1(:,i)$;
    $\boldsymbol{B}(:,k+2) = \boldsymbol{B}_2\boldsymbol{U}_2(:,i)$;
    $k = k + 2$;
  end if
end for
if $k_1 > k_{min}$ **then**
  $\boldsymbol{B}(:,k_{min}+1:k_1) = \boldsymbol{B}_1\boldsymbol{U}_1(:,k_{min}+1:k_1)$;
  $k = k + k_1 - k_{min}$;
else if $k_2 > k_{min}$ **then**
  $\boldsymbol{B}(:,k_{min}+1:k_2) = \boldsymbol{B}_2\boldsymbol{U}_2(:,k_{min}+1:k_2)$;
  $k = k + k_2 - k_{min}$;
end if

Output: Basis matrix $\boldsymbol{B} \in \mathbb{R}^{d\times k}$, the number of data N to generate $\boldsymbol{B}$

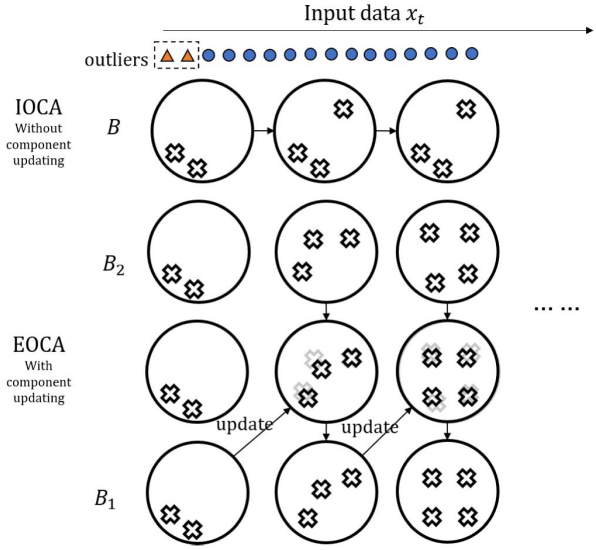


Fig. 3. Illustration for the comparison of IOCA and EOCA when outliers emerge at the beginning of learning. EOCA generates a subspace sequence, represented by basis set $\boldsymbol{B}_2^{(1)}, \boldsymbol{B}_2^{(2)}, \ldots, \boldsymbol{B}_2^{(i)}$ based on the input data stream. According to the abovementioned subspace sequence, EOCA executes the subspace updating operations as follows. At updating time t (at the end of learning period), it employs $\boldsymbol{B}_2^{(i)}$ to update $\boldsymbol{B}_1^{(i)}$'s basis. After updating $\boldsymbol{B}_1^{(i)}$, $\boldsymbol{B}_2^{(i)}$ will be cleaned, and the learning threshold will not be effected by the outliers. However, IOCA only updates one subspace and is more sensitive to the quality of samples at the beginning.

$\mathcal{S}^{(\text{new})}$. Considering that σ_i is rarely accurate 0 in the actual computation, we change the condition to $\sigma_i < 10^{-8}$. Note that we should orthonormalize the two vectors before adding them to the basis set, i.e., $\boldsymbol{w}_{2,i} = \boldsymbol{w}_{2,i} - \boldsymbol{w}_{1,i}\boldsymbol{w}_{1,i}^T\boldsymbol{w}_{2,i}$ and $\boldsymbol{w}_{2,i} = (\boldsymbol{w}_{2,i}/\|\boldsymbol{w}_{2,i}\|_2)$.

Moreover, the following rules are laid down for preserving information in $\mathcal{S}$ and $\mathcal{S}'$: if $k_1 > k_{min}$, then the $k_1 - k_{min}$ basis vectors $\boldsymbol{w}_{1,k_{min}+1}, \ldots, \boldsymbol{w}_{1,k_1}$ of $\mathcal{S}$ are directly added into the basis set of $\mathcal{S}^{(\text{new})}$; similarly, if $k_2 > k_{min}$, then the $k_2 - k_{min}$ basis vectors $\boldsymbol{w}_{2,k_{min}+1}, \ldots, \boldsymbol{w}_{2,k_2}$ of $\mathcal{S}'$ are directly added into the basis set of $\mathcal{S}^{(\text{new})}$.

Note that through (18) and (19), we know that when the first auxiliary subspace $\boldsymbol{B}_{21}$ is polluted by the initial noise, if N is larger, i.e., the (N_{21}/N) is smaller for $\boldsymbol{B}_{21}$, the noise will bring less effect to the following feature space. This point will also be analyzed in the experiment part.

In Algorithm 1, we give the process of orthonormal component updating. Given k_1-dimensional subspace $\mathcal{S}$ and k_2-dimensional subspace $\mathcal{S}'$ that are represented by basis matrix $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$, respectively, Algorithm 1 merges $\mathcal{S}$ and $\mathcal{S}'$ to obtain a new subspace $\mathcal{S}^{(\text{new})}$ that is represented by its basis matrix $\boldsymbol{B}$.

D. EOCA Framework

By combining online orthonormal component extraction and updating, we propose a novel subspace learning algorithm EOCA for linear dimensionality reduction.

EOCA incrementally learns feature subspace $\mathcal{S}$ that is represented by its orthonormal bases from the online data stream. During subspace learning, EOCA does not directly update $\mathcal{S}$ every time a new data comes in. Instead, it generates an auxiliary $\mathcal{S}'$ by continuously extracting OCs from the input data. Once $\mathcal{S}'$ achieves a steady state, it employs $\mathcal{S}'$ to update $\mathcal{S}$.

As illustrated in Fig. 3, the learning process of EOCA can be comprehended as follows. During the EOCA process, the online input data stream is incrementally learned, and it adaptively generates a series of subspaces. Thus, a subspace sequence $\mathcal{S}'_{(1)}, \mathcal{S}'_{(2)}, \ldots, \mathcal{S}'_{(i)}$ is obtained, which is represented by the basis sets $\boldsymbol{B}_2^{(1)}, \boldsymbol{B}_2^{(2)}, \ldots, \boldsymbol{B}_2^{(i)}$. Here, $\mathcal{S}'_{(i)}$ is the subspace learned from the ith data subsequence, and every subspace $\boldsymbol{B}_2^{(i)}$ is learned online. Then, EOCA employs the subspaces in the sequence one by one to update $\mathcal{S}$, which is represented by $\boldsymbol{B}_1^{(i)}$. The orthonormal bases of the final $\mathcal{S}$ are what we want.

In Algorithm 2, we present the detailed EOCA algorithm. In the beginning, both $\mathcal{S}$ and $\mathcal{S}'$ are initialized as 0-D subspaces. In the learning process, EOCA continuously extracts new OCs and adds them into the basis set of $\mathcal{S}'$. If $\mathcal{S}'$ remains unchanged, i.e., its dimension does not increase, we deem that $\mathcal{S}'$ achieves a steady state after learning t_0 samples. Then, we employ $\mathcal{S}'$ to update $\mathcal{S}$ by Algorithm 1. After the orthonormal component updating, $\mathcal{S}'$ is initialized as 0-D subspace again, and EOCA continues to extract OCs for $\mathcal{S}'$. When all the data have been processed, if the dimension of $\mathcal{S}'$ is larger than 0, EOCA performs component updating in the end. In Algorithm 2, h is the times of subspace basis updating; t_0 is the parameter to judge whether the auxiliary $\mathcal{S}'$ achieves a steady state. In this article, the default setting is $t_0 = d$.

Algorithm 2 Algorithm of EOCA

Input: Data $\boldsymbol{x}_1 \ldots, \boldsymbol{x}_N$ ($\boldsymbol{x}_t \in \mathbb{R}^d,\ t = 1, \ldots, N$), parameter t_0

Initialize $\boldsymbol{B}_1 = [\ ]$, $\boldsymbol{B}_2 = [\ ]$; Let $N_1 = 0$, $N_2 = 0$, $k_2 = 0$, $t' = 0$, $L_{\max} = 0$, $h = 0$;
for $t = 1, \ldots, N$ **do**
  Input $\boldsymbol{x}_t \in \mathbb{R}^d$;
  if $t - t' > t_0$ **then**
    if $N_1 == 0$ **then**
      Let $\boldsymbol{B}_1 = \boldsymbol{B}_2$, $N_1 = N_2$;
    else
      Input $\boldsymbol{B}_1$, $\boldsymbol{B}_2$, N_1 and N_2 to Algorithm 1, obtain the updated $\boldsymbol{B}_1$ and N_1;
    end if
    Let $h = h+1$, $\boldsymbol{B}_2 = [\ ]$, $N_2 = 0$, $k_2 = 0$ and $L_{\max} = 0$;
  end if
  Update $N_2 = N_2 + 1$;
  if $\|\boldsymbol{x}_t\|_2 > L_{\max}$ **then**
    $L_{\max} = \|\boldsymbol{x}_t\|_2$;
  end if
  Let $\boldsymbol{r}_t = \boldsymbol{x}_t$;
  Compute $\boldsymbol{r}_t = \boldsymbol{r}_t - \sum_{i=1}^{k_2} \boldsymbol{b}_i \boldsymbol{r}_t{}^\top \boldsymbol{b}_i$;
  Compute $\boldsymbol{b}_{\boldsymbol{k_2+1}} = \frac{\boldsymbol{r}_t}{\|\boldsymbol{r}_t\|_2}$;
  if $\frac{\|\boldsymbol{r}_t\|_2}{L_{\max}} \geq f\left(\frac{k_2}{d}\right)$ **then**
    Let $\boldsymbol{B}_2 = [\boldsymbol{B}_2, \boldsymbol{b}_{\boldsymbol{k_2+1}}]$, $k_2 = k_2 + 1$, $t' = t$;
  end if
end for
Input $\boldsymbol{B}_1$, $\boldsymbol{B}_2$, N_1 and N_2 to Algorithm 1, obtain the updated $\boldsymbol{B}_1$ and N_1;

Output: Basis matrix $\boldsymbol{B}_1$

IV. Algorithm Analysis

A. Uniqueness of Linear Subspace

The linear subspace $\mathcal{S}^{(\text{new})}$ is combined with two learned subspaces $\mathcal{S}$ and $\mathcal{S}'$. We know that the results of SVD, which is a critical step of subspace updating, are not unique. However, we can claim that, no matter which result is obtained, the updating result $\mathcal{S}^{(\text{new})}$ is unique when two subspaces $\mathcal{S}$ and $\mathcal{S}'$ are given. The uniqueness of $\mathcal{S}^{(\text{new})}$ is analyzed in the Supplementary Material. Due to this conclusion, Algorithm 1 can be safely employed to update the orthonormal components and learn the subspace sequence in online environments.

B. Analysis on Computational Complexity

The computational complexity of EOCA depends on two aspects: component extraction and component updating.

Suppose that k is the final dimension of feature subspace, and N is the number of input data. For component extraction, the calculation of computing $\boldsymbol{r}_t$ takes $O(dk)$ for the size of basis set is $O(k)$, and the dimension is $O(d)$; for the whole N samples, this method takes $O(Ndk)$ time.

For component updating, computational complexity of Algorithm 1 is $O(dk^2)$ for the computation of SVD. Note that given $\mathcal{S}$ and $\mathcal{S}'$, k-dimensional subspace of the newly obtained $\mathcal{S}^{(\text{new})}$ satisfies $\max\{k_1, k_2\} \leq k \leq k_1 + k_2$, and it achieves its upper limit when $\mathcal{S}'$ is perpendicular to $\mathcal{S}$. In Algorithm 1, the computational complexity of SVD is $O(dk^2)$, and the linear transformation in basis alignment and adjustment takes $O(dk^2)$ time. Therefore, the proposed orthonormal component updating algorithm's computational complexity is $O(dk^2)$. In the learning process, the updating process is conducted $O(N/t_0)$ times. If we use the default setting of parameter $t_0 = d$, the global computation on component updating is $O(Nk^2)$. As a result of that, the total computational complexity of EOCA is $O(Ndk) + O(Nk^2) = O(Ndk)$.

V. Experiments

To evaluate the performance of EOCA, we conduct experiments on three synthetic data sets and 13 real-world data sets. All the experiments in this article are performed in MATLAB R2019a on an Ubuntu server.

A. Experiments on Synthetic Data

In this section, we take experiments on synthetic data sets to evaluate the ability of EOCA on target dimensionality estimation and dealing with outliers. We use IOCA to compare their performance on noisy data sets since IOCA lacks the ability to update the basis set. The synthetic data are generated as follows. First, d_0 mutually orthonormal basis vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{d_0} \in \mathbb{R}^d$ are randomly generated. Then, N data $\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_N$ are obtained by the linear combination of these d_0 vectors, while the combination coefficients follow standard normal distribution. Assume that $\tilde{x}_{i,j}$ to be the jth entry of $\tilde{\boldsymbol{x}}_i$ and $x_m = (1/dN)\sum_{i,j} |\tilde{x}_{i,j}|$, and we compute the data with noise by adding Gaussian noise proportional to x_m

$$\boldsymbol{x}_i = \tilde{\boldsymbol{x}}_i + 0.02 \cdot \text{randn}(d, 1) \cdot x_m. \tag{21}$$

Moreover, we generate an extra vector as outlier by

$$\boldsymbol{x}_0 = \lambda \|\text{randn}(d, 1)\|_2 \boldsymbol{w}_0 \tag{22}$$

where $\boldsymbol{w}_0$ is a basis that is orthogonal to S_0, and λ is a parameter that varies in the experiments. Thus, we obtain a data set containing $N + 1$ samples.

Suppose that EOCA is employed on the synthetic data, it obtains a k-dimensional subspace $\mathcal{S} = \text{span}\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\}$, and the directional distance from $\mathcal{S}_0 = \text{span}\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{d_0}\}$ to $\mathcal{S}$ can be measured as follows [40]:

$$\text{dist}(\mathcal{S}_0, \mathcal{S}) = \sqrt{d_0 - \sum_{i=1}^{\min(d_0,k)} \cos^2 \theta_i} \tag{23}$$

$$= \sqrt{d_0 - \sum_{i=1}^{d_0} \sum_{j=1}^{k} \left(\boldsymbol{w}_i^\top \boldsymbol{b}_j\right)^2}. \tag{24}$$

Thus, when d is fixed, the smaller $\text{dist}^2(\mathcal{S}_0, \mathcal{S})$ is, the better the quality of the learned $\mathcal{S}$ is. For convenience, we employ $\text{dist}^2(\mathcal{S}_0, \mathcal{S})$ to measure the similarity between $\mathcal{S}_0$ and $\mathcal{S}$.

Besides IOCA, we also implemented a baseline method named evolutionary non-OC analysis (ENCA), which is almost the same as EOCA except employing nonorthogonal bases. This is to show the impact of orthogonal bases compared

TABLE I

RESULTS OF EOCA SELECTING THRESHOLD FUNCTION ON SYNTHETIC DATA WHEN $N = 200$

		$f(\omega) = \sqrt{\omega}$			$f(\omega) = \omega$			$f(\omega) = \omega^2$			ENCA ($f(\omega) = \omega$)			
	λ	k	$dist^2(S_0, S)$	h	k	$dist^2(S_0, S)$	h	k	$dist^2(S_0, S)$	h	k	$dist^2(S_0, S)$	h	$\|\|\boldsymbol{I} - \boldsymbol{B}_1^T \boldsymbol{B}_1\|\|_2$
$d_0 = 10$ $d = 30$	–	9.74	0.26	4.26	10	3.9×10^{-3}	4.94	10.86	4.1×10^{-3}	4.96	10	7.6×10^{-3}	4.94	0.08
	2	9.7	0.58	4.2	10.3	0.090	4.7	11.1	4.5×10^{-3}	5	10.3	0.13	4.7	0.09
	3	9.8	0.39	4.3	10.1	0.19	4.9	11.1	4.7×10^{-3}	5	10.1	0.23	4.9	0.09
	5	9.6	0.51	4.6	10	0.21	5	11.1	0.025	5	10	0.75	5	0.10
	10	9.7	0.36	4.6	10	0.10	5	10.9	0.058	5	10	0.54	5	0.10
$d_0 = 10$ $d = 30$	–	9.74	0.26	4.26	10	3.9×10^{-3}	4.94	10.86	4.1×10^{-3}	4.96	10	7.1×10^{-3}	4.94	1.09
	2	10.9	0.11	2	11.2	0.011	2	16.6	1.8×10^{-3}	2	11.2	0.034	2	0.92
	3	10.1	0.86	1.8	11	0.014	2	16.4	2.3×10^{-3}	2	11	0.037	2	0.94
	5	8.4	2.52	1.9	11	8.5×10^{-3}	2	16.2	3.0×10^{-3}	2	11	8.8×10^{-3}	2	0.98
	10	9.5	1.33	1.9	11	8.4×10^{-3}	2	16	5.3×10^{-3}	2	11	0.011	2	1.06
$d_0 = 30$ $d = 100$	–	26.44	3.58	1.56	30.82	0.060	1.91	32.2	0.042	1.93	30.82	1.57	1.91	1.91
	2	20.9	10.11	1.4	30.5	1.43	1.7	32.1	0.062	2	30.5	1.76	1.7	1.85
	3	22.2	8.71	1.3	30.1	3.80	1.4	31.9	0.065	2	30.1	6.12	1.4	1.80
	5	22.4	8.46	2	29.8	5.83	1.4	31.88	0.15	1.9	29.8	7.09	1.4	1.72
	10	25.3	5.34	2	30.2	4.80	1.8	31.44	2.92	1.6	30.2	6.73	1.8	1.74

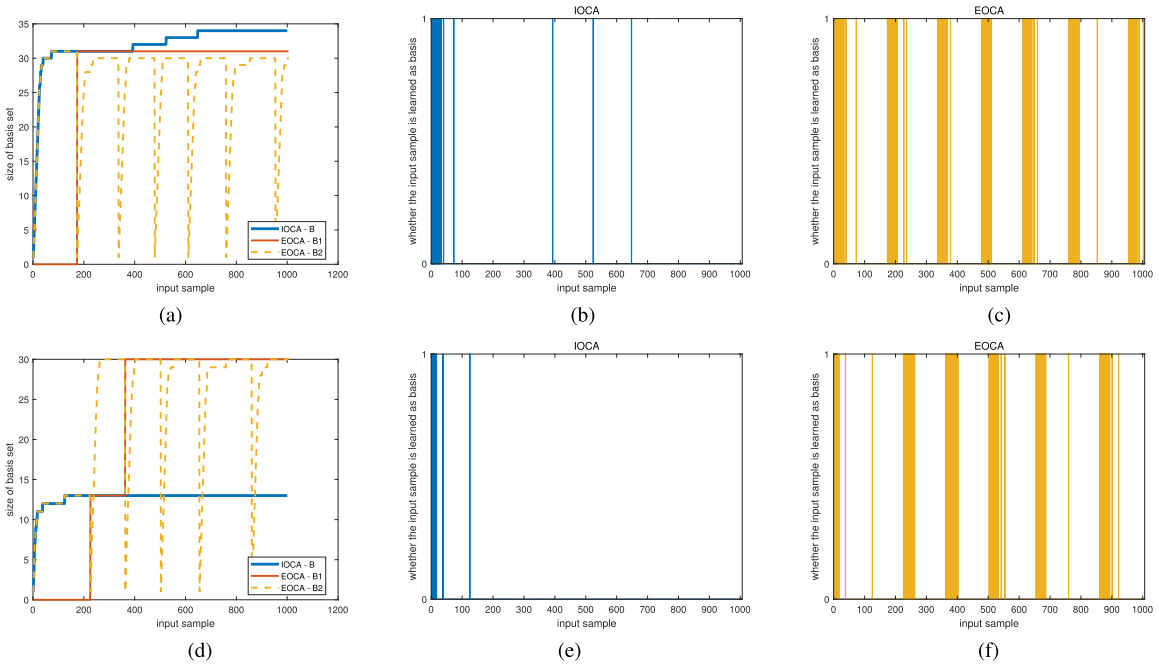


Fig. 4. Size of basis set changing with input samples for IOCA and EOCA on the synthetic noisy data when $N = 1000$, $d_0 = 30$, and $d = 100$. Five extra outliers generated as x_o and Gaussian noise are added on data set. (a)–(c) $\lambda = 2$. (d)–(f) $\lambda = 10$. $\boldsymbol{B}$, $\boldsymbol{B}_1$, and $\boldsymbol{B}_2$ are basis matrixes in IOCA and EOCA learning process. The size of $\boldsymbol{B}$ and $\boldsymbol{B}_1$ also represents the dimensionality of subspaces that are learned for reduction. The first column represents the change of basis set size for IOCA and EOCA, and the lines in the second and third columns represent that the corresponding sample is added to basis set. (a) $\lambda = 2$. (b) Input samples added to B, IOCA. (c) Input samples added to B_2, EOCA. (d) $\lambda = 10$. (e) Input samples added to B, IOCA. (f) Input samples added to B_2, EOCA.

with nonorthogonal ones. In this method, we regard the GS procedure as computation of adaptive threshold. As proven in [41], the GS procedure is a solution for least-squares problem, i.e., $\min_{\alpha_i,t,\ldots,\alpha_{k_2},t} \| \sum_{i=1}^{k_2} \alpha_{i,t} \boldsymbol{b}_i - \boldsymbol{x}_t \|_2$. In other words, if we compute the coefficients using the least-squares equation and then compute the residual vector, the result is the same as using the GS procedure. More details about this result are discussed in [35]. As a result, we can similarly compute coefficients with the minimum norm least-squares solution

$$\boldsymbol{\alpha}_t^* = (\boldsymbol{B}^T \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{x}_t \tag{25}$$

$$\boldsymbol{r}_t = \boldsymbol{B}_2 \boldsymbol{\alpha}_t^* - \boldsymbol{x}_t \tag{26}$$

to replace the step 17 in Algorithm 2 and the step 20 in Algorithm 2 with $\boldsymbol{B}_2 = [\boldsymbol{B}_2, (\boldsymbol{x}_t / \|\boldsymbol{x}_t\|_2)]$ to compute nonorthogonal bases. The baseline method keeps the adaptive threshold computation, and all the other operations are unchanged in EOCA.

Three results are presented on different settings. First, we compare and show different results of EOCA when selecting different threshold functions, i.e., $f(\omega) = \sqrt{\omega}$, $f(\omega) = \omega$, and $f(\omega) = \omega^2$, on the setting that $N = 200$. Moreover, we report the results of the baseline ENCA with threshold $f(\omega) = \omega$. The nonzero values of $\|\boldsymbol{I} - \boldsymbol{B}_1^T \boldsymbol{B}_1\|_2$ show that those bases learned by ENCA are actually nonorthogonal. The result is reported in Table I. Then, we compare EOCA with

TABLE II
RESULTS OF EOCA AND IOCA ON SYNTHETIC DATA WHEN $N = 2000$

		EOCA			IOCA	
	λ	k	$dist^2(S_0, S)$	h	k	$dist^2(S_0, S)$
$d_0 = 10$ $d = 30$	–	10	3.8×10^{-4}	43.67	10	0.017
	2	10.5	1.2×10^{-3}	43.5	11	0.0081
	3	10	2.2×10^{-3}	43.3	10.3	0.70
	5	10	2.3×10^{-3}	43.4	8.9	2.10
	10	10	1.8×10^{-3}	43.8	5.6	5.40
$d_0 = 10$ $d = 100$	–	11.4	1.0×10^{-3}	16.31	10.7	0.017
	2	11.9	1.1×10^{-3}	16.4	11.2	0.023
	3	11.8	1.2×10^{-3}	16.4	11.2	0.020
	5	11.9	1.0×10^{-3}	16.3	11	0.017
	10	11.5	3.0×10^{-3}	16	10.2	0.80
$d_0 = 30$ $d = 100$	–	30	6.1×10^{-3}	11.98	30	0.075
	2	30.1	0.028	12	30.6	0.43
	3	30	0.044	11.8	28.3	2.72
	5	30	0.027	12	21.3	9.71
	10	30	0.027	11.9	12.6	18.40

IOCA [34] by employing IOCA and EOCA on the previously generated data for $N = 2000$ with one outlier added at the beginning, and the threshold is chosen as $f(\omega) = \omega$. The result is reported in Table II. Finally, we visualize the learning process when $N = 1000$, $f(\omega) = \omega^2$, $d_0 = 30$, and $d = 100$ and add five outliers in the beginning. The results are reported in Fig. 4. When λ is noted as $-$, the N normal data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ are learned in random order, but the outlier $\boldsymbol{x}_0$ is not input. We report the mean results of 100 runs.

When $\lambda = 1$, the outlier $\boldsymbol{x}_0$ can be seen as normal input. When $\lambda > 1$, outliers first come in, and then, the other N data are fed in random order. h is the times of subspace merging.

From Table I, we find that the threshold functions $f(\omega) = \sqrt{\omega}$ and $f(\omega) = \omega$ often lead to a lower dimension estimation and $f(\omega) = \omega^2$ leads to a higher dimension estimation. This result is similar to the experiment in [34]. On the whole, the threshold function $f(\omega) = \omega$ performs better. However, when $d_0 = 10$ and $d = 100$, the higher dimension estimation of $f(\omega) = \omega^2$ is more close to the ideal target. We can find that $f(\omega) = \omega^2$ provides a more loose threshold, and EOCA learns more bases in the learning process. The situation is contrary for the function $f(\omega) = \sqrt{\omega}$. On the whole, we believe that we can choose $f(\omega) = \omega$ in all situations. On the other hand, we can see that ENCA performs similarly on some results (k, h) but worse than EOCA on the generation of subspaces. They form a worse embedded space $\mathcal{S}$ than the normal EOCA procedure evaluated by $\text{dist}^2(\mathcal{S}_0, \mathcal{S})$. For orthogonality, the values of $\|\boldsymbol{I} - \boldsymbol{B}_1^T \boldsymbol{B}_1\|_2$ computed by the bases obtained in EOCA are about 10^{-15}, so the bases can be regarded as orthogonal, while the bases of ENCA is completely nonorthogonal according to the results.

From Table II, we can find that when λ is large, the outlier has less influence on the performance of EOCA than that of IOCA. IOCA cannot adjust the extracted component during learning. Due to the threshold policy in component extraction, if λ is large, when $\boldsymbol{x}_0$ has been processed, the threshold becomes too strict ($L_{\max}$ is large). Combined with Fig. 4, we can see that IOCA stops learning a new basis with outliers processed when $\lambda = 10$. However, when $\lambda = 2$, we can see that the number of subspaces IOCA learned is larger than the EOCA learned and the ground truth. It indicates that IOCA regards all the outliers as normal data. Therefore, outlier $\boldsymbol{x}_0$ at first is the worst case for IOCA. On the other hand, EOCA is able to update the extracted component, and the target dimensionality k determined by EOCA is more coincident with d_0 being the intrinsic dimension of the data set without $\boldsymbol{x}_0$. Meanwhile, the $\text{dist}^2(\mathcal{S}_0, \mathcal{S})$ obtained by EOCA is smaller than IOCA.

Comparing the results of EOCA in Tables I and II, we can find that EOCA generates a better basis set with outliers when more data are input and the times of subspace updating increase. Also, we can find that the more EOCA conducts component updating operation in the learning process, the less effect of result that is caused by outliers. We already know that updating operation is conducted at most (N/d) times. Intuitively, when the number of data increases, EOCA will perform more subspace updating operations and get better results. Thus, we can conclude that EOCA performs better with more data when outliers exist in the data set.

From Fig. 4, we can explore more on why subspace updating reduces the effect of outliers. In this figure, we show the changing size of the basis set and whether the input sample is learned as a new basis. The input samples added to the first subspace $\boldsymbol{B}_2$ in EOCA are the same; namely, the first learning stage of EOCA is, indeed, the same with $\boldsymbol{B}$ in IOCA. However, when $\boldsymbol{B}_2$ reaches a steady state, the subspace updating process will be conducted. Thus, if there is enough data, more subspaces $\boldsymbol{B}_2$ will be generated during the learning process of EOCA. After subspace updating, the learned parameters of $\boldsymbol{B}_2$ are cleared, and the outliers, in the beginning, will not affect the following adaptive threshold of $\boldsymbol{B}_2$. In other words, they will only affect the first subspaces where they are learned, and the following subspaces will be learned normally. The number of outliers should be small and can only affect a small number of subspaces. In a series of subspace adjustments, it is hard for the small number of subspaces that are distorted by the outlier to pollute the final result, and more combination with normal subspaces will reduce the effect of polluted subspace. Thus, compared with IOCA, EOCA is more robust on noisy data set, especially when the outliers emerge at the beginning of the data set. In summary, the component updating method described in Algorithm 1 plays a key role in reducing the effect of outliers. The comparison is also illustrated in Fig. 3.

We can draw the conclusion that, in the initial outlier case, compared with IOCA, EOCA achieves better performance with the help of the proposed online component updating strategy.

B. Experiments on Real-Word Data

1) Noise-Free Data Sets: First, we compare EOCA with incremental dimensionality reduction algorithms IOCA [34], IPCA [49], CCIPCA [50], SGA [51], GHA [52], and nonlinear PCA (nPCA) [53] in online environment. Also, the neural network denoising autoencoder (DAE) is also compared with a batch learning benchmark. In this experiment, we try to

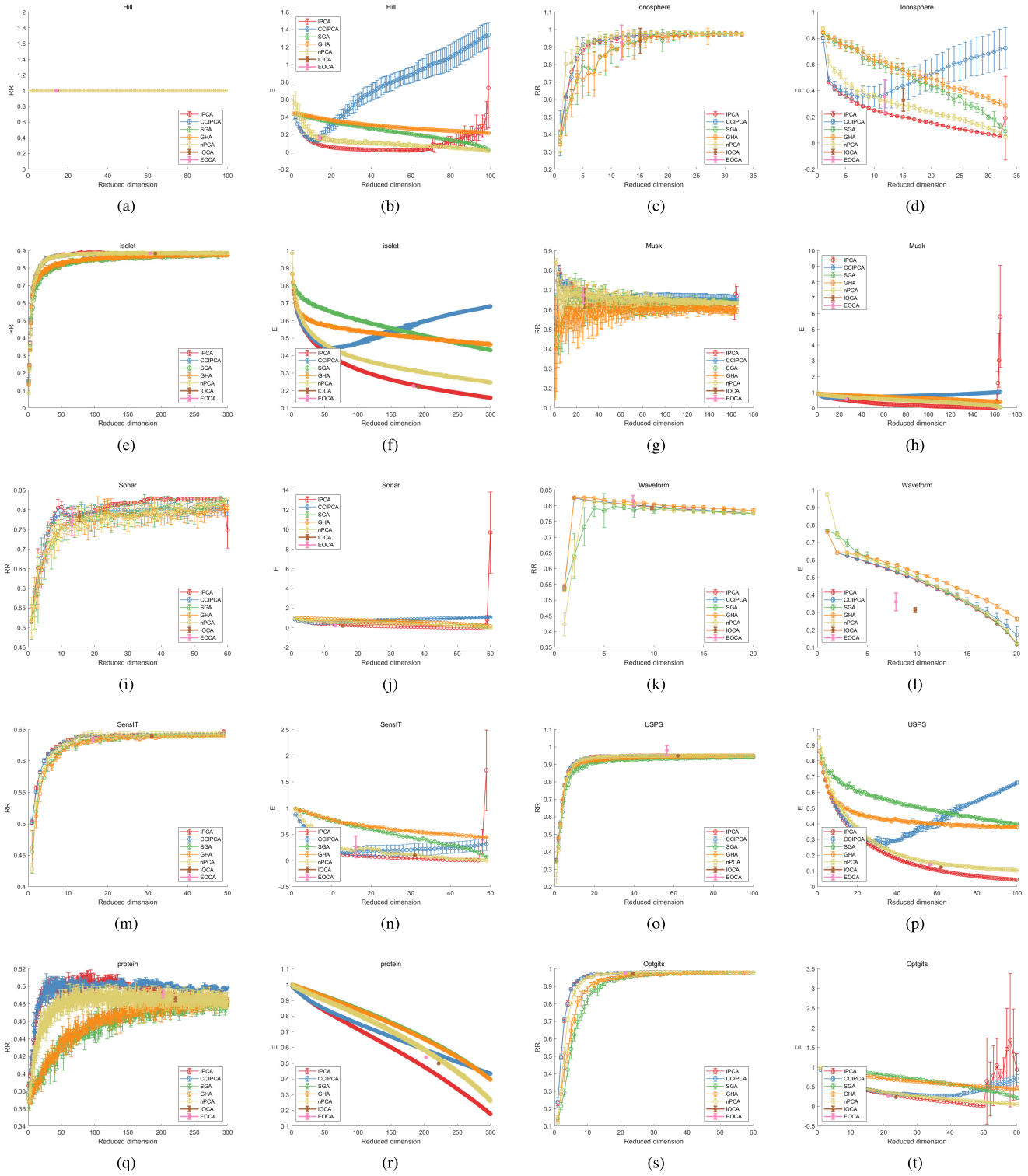


Fig. 5. Results on nonnoisy data sets of comparing method. Two metrics are compared, including RR and reconstruction error (E). The bar indicates the mean and standard deviation. (a) Hill, RR. (b) Hill, E. (c) Ionosphere, RR. (d) Ionosphere, E. (e) ISOLET, RR. (f) ISOLET, E. (g) Musk, RR. (h) Musk, E. (i) Sonar, RR. (j) Sonar, E. (k) Waveform, RR. (l) Waveform, E. (m) SensIT Vehicle, RR. (n) SensIT Vehicle, E. (o) USPS, RR. (p) USPS, E. (q) protein, RR. (r) protein, E. (s) Optdigits, RR. (t) Optdigits, E.

validate the performance of EOCA in the common circumstances (without outliers). The reported results of these algorithms are their average results obtained after ten executions, in which the input order of the data sequence is randomly generated. We compare the performance of these algorithms on 13 real-world data sets from several application fields and DAE on three data sets, including MNIST, IJCNN1, and cifar10. The details of these data sets are shown in Table III.

TABLE III
DETAILS OF REAL-WORLD DATA SETS

Dataset	#dimensionality	#train	# test	#label
Hill [42]	100	606	606	2
Ionosphere [42]	34	251	100	2
Musk [42]	166	476	6598	2
OptDigit [42]	64	3823	1797	10
Sonar [42]	60	104	104	2
Waveform [42]	21	900	4100	3
IJCNN1 [43]	22	49990	91701	2
ISOLET [42]	617	6238	1559	26
MNIST [44]	784	60000	10000	10
protein [45]	357	17766	6621	3
SensIT Vehicle [46]	50	78823	19705	3
USPS [47]	256	7291	2007	10
cifar10 [48]	3072	50000	10000	10

After all the original data are transformed into low-dimensional feature subspace learned from the training set, the one-nearest neighbor classifier is employed for incremental methods to classify the testing data. DAE employs softmax classification layer and Adam optimizer with learning rate of 0.001. DAE is first trained on overall reconstruction error for hidden layers and then fine-tuned on classification error for softmax and hidden layers. It is trained with masking-noise, with corruption levels 30%. For nPCA, we use $g(t) = sgn(t)ln(1+\alpha(t))$ and $\alpha = 1$. For SGA and GHA, the parameters are set as $\alpha = 0.7$ and $c = 0.05$. Two metrics are exploited in the experiments, i.e., recognition rate RR and mean relative reconstruction error E. RR is computed as the classification accuracy of one-nearest neighbor classifier to evaluate whether the learned components can be classified easily, and E is computed as

$$E = \frac{1}{N}\sum_{t=1}^{N}\frac{\left\|\boldsymbol{x}_t - \sum_{j=1}^{k}\boldsymbol{b}_j\boldsymbol{b}_j^{\top}\boldsymbol{x}_t\right\|_2}{\|\boldsymbol{x}_t\|_2} \tag{27}$$

to testify whether the learned components can accurately approximate the original data. E is computed before DAE fine-tune with labels. The MNIST, Optdigits, Hill, IJCNN1, and cifar10 data sets are normalized, and the others are remained unchanged.

The experimental results are summarized in Fig. 5 and Table IV. For the recognition rate, EOCA achieves competitive results with comparing methods. We can conclude that, on normal data sets, EOCA performs similar to incremental methods with the data set not contaminated with noisy data. DAE gets high RR but similar reconstruction error with other incremental methods. From the result, we can also find that the results on two metrics are not always consistent, i.e., the best reconstruction of original data does not represent the best classification accuracy with a 1-NN classifier. Also, EOCA and IOCA perform similarly on the noise-free data sets.

2) Noisy Data Sets: To demonstrate the robustness of EOCA, we add outliers into the real-world data sets and take experiments again. On the noisy data set, we compare four linear methods: IPCA, CCIPCA, IOCA, and EOCA. SGA and GHA are sensitive to the noise that has a big norm, so they behave not so good on the data sets with this noise type. Given a data set $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, we let

$$L = \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{x}_i\|_2 \tag{28}$$

and each outlier is generated in the form of $20L\boldsymbol{z}$; here, $\boldsymbol{z}$ is a randomly generated unit vector. Their labels are randomly picked from the original data set. To demonstrate the methods on different noisy environments, we define noise rate C. For each data set, $\lfloor CN \rfloor$ outliers are generated, and they are fed before the normal data. Different from the nonnoisy environment, we do not shuffle the data sets in order to demonstrate their performance on different noisy data. EOCA, IOCA, IPCA, and CCIPCA are compared on these data sets, and the target dimensionality k determined by EOCA is employed by IPCA, CCIPCA, and nPCA. Table V reports the average results of ten executions.

Comparing Table V with nonnoisy results, EOCA gets good results and performs stable with noisy data. With more noise, EOCA ranks better (from 16/24 best, i.e., it ranks best on 16 results out of all 24 results, to 20/24 best). On most data sets, EOCA exceeds IOCA on both two metrics. We can find that the numbers of OCs extracted by IOCA vary with the noisy environment. In some settings, the dimension decreases significantly. However, the dimension increases on other data sets. This result is in accordance with that of synthetic data with noise.

For EOCA, the outlier in which ℓ_2-norm is large pushes it to begin the next round of subspace generation earlier. Through subspace merging, the extracted OCs are updated, and the influence of the outliers is reduced. We can find that the results of EOCA in Table V are similar (comparing different C and normal data sets), while comparing methods perform worse with more noise. Therefore, we believe that the proposed OC updating strategy is effective in eliminating the influence of outliers.

In summary, EOCA achieves competitive results with several typical incremental dimensionality reduction methods on the normal data set. Moreover, EOCA behaves more robustly on data sets with outliers compared with the previous component learning method IOCA (which only employs component extraction method), IPCA, and CCIPCA. It shows that the component updating process is effective in dealing with noisy data in online data sets. Thus, we can conclude that EOCA is a good choice for online data processing when outliers or noisy data emerge at first in the data set.

3) Computational Complexity Comparison: As low time cost is an excellent property for online learning, we compare both the theoretical computational complexity and the actual runtime for these comparing algorithms. The comparing results are presented in Table VI. Although most of the fast methods are the same complexity $O(Ndk)$, the constant C before the equation, the implement details, and the machine state will influence the actual runtime. For nPCA, the matrix multiplication will consume $O(Ndk^2)$, while the complexity of $g(t)$ is determined by the function. The actual runtime is recorded on the noise-free data set using MATLAB tic/toc command. We can see that, on the whole, the speed of these methods

TABLE IV

RESULTS COMPARING DAE AND OTHER INCREMENTAL METHODS ON NORMAL DATA SET. IF DAE IS THE BEST RESULT, IT IS IN BOLD

data set	metric	EOCA	IPCA	CCIPCA	SGA	GHA	nPCA	IOCA	DAE
IJCNN1	k	12.0	12.0	12.0	12.0	12.0	12.0	12.0	16.0
	RR(%)	93.71 ± .34	95.40 ± .13	93.12 ± .00	100 ± .00	95.88 ± 1.66	93.95 ± 1.79	95.63 ± 1.45	**97.63 ± 0.02**
	E	.389 ± .058	.281 ± .012	.292 ± .034	.529 ± .016	.479 ± .044	.275 ± .031	.282 ± .103	.317 ± .003
MNIST	k	192.0	192.0	192.0	192.0	192.0	192.0	193.5	192.0
	RR(%)	96.94 ± .10	96.89 ± .08	96.93 ± .12	96.23 ± .09	96.24 ± .07	96.98 ± .11	96.95 ± .09	**97.30 ± .01**
	E	.243 ± .015	.246 ± .017	.470 ± .022	.503 ± .022	.512 ± .033	.354 ± .002	.247 ± .008	**.242 ± .002**
cifar10	k	341.0	341.0	341.0	341.0	341.0	341.0	392.0	341.0
	RR(%)	37.15 ± .24	36.67 ± .13	36.36 ± .31	33.42 ± .16	34.26 ± .29	38.38 ± .23	37.23 ± .12	**39.41 ± .03**
	E	.126 ± .035	.181 ± .020	.317 ± .024	.441 ± .102	.445 ± .088	.437 ± .023	.120 ± .027	.210 ± .002

TABLE V

PERFORMANCES OF EOCA AND COMPARING METHODS ON THE DATA SETS WITH OUTLIERS OF DIFFERENT NOISE RATE C. THE RESULT IS SHOWN AS $mean \pm std$ AND THE BEST IN BOLD. IF THE PERFORMANCE OF EOCA IS BETTER THAN IOCA, IT WILL BE UNDERLINED. (a) $C = 0.01$. (b) $C = 0.1$

data set	metric	EOCA	IPCA	CCIPCA	IOCA	data set	metric	EOCA	IPCA	CCIPCA	IOCA
Hill	k	13.0	13.0	13.0	9.0	IJCNN1	k	10.0	10.0	10.0	12.6
	RR(%)	100 ± .00	100 ± .00	100 ± .00	100 ± .00		RR(%)	91.63 ± 0.00	**95.88 ± 1.66**	93.95 ± 1.79	95.63 ± 1.45
	E	**.171 ± .000**	.173 ± .002	.280 ± .025	.354 ± .002		E	**.350 ± .000**	.603 ± .022	.626 ± .019	.537 ± .020
Ionosphere	k	12.0	12.0	12.0	8.0	ISOLET	k	147.0	147.0	147.0	65.0
	RR(%)	92.00 ± .00	**94.90 ± 1.50**	94.00 ± 1.56	78.90 ± 1.10		RR(%)	**88.71 ± .00**	88.68 ± .18	86.40 ± .40	80.12 ± .11
	E	.415 ± .000	**.221 ± .014**	.372 ± .083	.525 ± .012		E	**.272 ± .000**	.313 ± .001	1.926 ± .091	.679 ± .003
Musk	k	29.0	29.0	29.0	7.0	cifar10	k	341.0	341.0	341.0	503.0
	RR(%)	**67.38 ± .00**	62.66 ± .60	64.63 ± 5.02	65.97 ± 9.04		RR(%)	37.23 ± .13	36.24 ± .21	**38.85 ± .05**	32.38 ± .44
	E	**.517 ± .000**	.525 ± .010	.853 ± .002	.832 ± .010		E	**.122 ± .002**	.521 ± .001	2.714 ± .070	.355 ± .008
OptDigits	k	19.0	19.0	19.0	28.5	protein	k	206.0	206.0	206.0	146.0
	RR(%)	**97.55 ± .00**	95.76 ± .50	–	96.28 ± .39		RR(%)	47.47 ± .00	49.27 ± .53	–	**46.55 ± .62**
	E	**.297 ± .000**	.679 ± .012	–	.444 ± .013		E	.549 ± .000	**.525 ± .001**	–	.726 ± .002
Sonar	k	16.0	16.0	16.0	8.0	SensIT Vehicle	k	22.0	22.0	22.0	25.0
	RR(%)	78.84 ± .00	**79.71 ± .84**	78.90 ± 1.94	71.87 ± 2.18		RR(%)	**63.89 ± .00**	63.50 ± .23	63.30 ± .19	63.51 ± .18
	E	.158 ± .000	**.152 ± .002**	.385 ± .06	.342 ± .003		E	**.126 ± .000**	.362 ± .013	.774 ± .052	.519 ± .016
Waveform	k	8.0	8.0	8.0	9.7	USPS	k	54.0	54.0	54.0	75.0
	RR(%)	**79.49 ± .00**	75.03 ± 3.79	74.22 ± .61	72.63 ± 3.01		RR(%)	**95.12 ± .00**	93.48 ± .19	93.92 ± .26	93.57 ± .35
	E	**.343 ± .000**	.603 ± .045	.892 ± .088	.403 ± .020		E	**.143 ± .000**	.510 ± .004	1.435 ± .073	.533 ± .004

(a) $C = 0.01$

data set	metric	EOCA	IPCA	CCIPCA	IOCA	data set	metric	EOCA	IPCA	CCIPCA	IOCA
Hill	k	15.0	15.0	15.0	43.3	IJCNN1	k	10.0	10.0	10.0	13.2
	RR(%)	100 ± .0	100 ± .0	100 ± .0	100 ± .0		RR(%)	93.45 ± .02	**95.60 ± .79**	95.30 ± 1.0	95.2 ± 1.1
	E	**.146 ± .008**	.255 ± .004	.262 ± .014	.340 ± .002		E	**.350 ± .000**	.379 ± .079	.810 ± .013	.527 ± .018
Ionosphere	k	12.0	12.0	12.0	16.4	ISOLET	k	167.0	167.0	167.0	256.0
	RR(%)	92.00 ± .00	89.90 ± 8.24	**92.90 ± 5.80**	91.30 ± 6.09		RR(%)	**88.54 ± .20**	86.72 ± .51	84.93 ± .47	87.68 ± .51
	E	.415 ± .000	**.2420 ± .016**	.255 ± .017	.443 ± .019		E	**.272 ± .000**	.313 ± .001	1.926 ± .091	.679 ± .003
Musk	k	29.0	29.0	29.0	50	cifar10	k	366.6	367	367	1239.2
	RR(%)	**67.36 ± .04**	66.99 ± 8.5	66.59 ± 6.4	59.42 ± 8.2		RR(%)	**37.15 ± .10**	33.89 ± .16	34.09 ± .19	34.37 ± .31
	E	**.517 ± .000**	.838 ± .005	.879 ± .045	.659 ± .008		E	**.122 ± .001**	.201 ± .001	.438 ± .023	.396 ± .004
OptDigits	k	18.0	18.0	18.0	31.0	protein	k	206.0	206.0	206.0	153.8
	RR(%)	**97.55 ± .00**	93.90 ± .5	–	96.69 ± .4		RR(%)	**47.53 ± .0**	47.52 ± .55	–	46.86 ± .88
	E	**.297 ± .000**	.300 ± .049	–	.436 ± .011		E	.549 ± .000	**.533 ± .001**	–	.713 ± .002
Sonar	k	16.0	16.0	16.0	13.0	SensIT Vehicle	k	22.0	22.0	22.0	26.6
	RR(%)	**78.84 ± .00**	76.63 ± 1.11	75.29 ± 1.70	76.64 ± 2.61		RR(%)	**63.89 ± .00**	63.57 ± .20	63.53 ± .35	63.63 ± .23
	E	**.158 ± .000**	.309 ± .010	.792 ± .096	.513 ± .019		E	**.126 ± .000**	.362 ± .013	.774 ± .052	.519 ± .016
Waveform	k	8.0	8.0	8.0	11.6	USPS	k	54.0	54.0	54.0	111.6
	RR(%)	**79.49 ± .00**	69.58 ± 3.4	70.54 ± 2.8	74.12 ± 3.0		RR(%)	**95.07 ± .11**	93.15 ± .44	92.81 ± .34	94.22 ± .24
	E	**.343 ± .000**	.403 ± .014	.544 ± .031	.376 ± .021		E	**.142 ± .003**	.287 ± .029	.311 ± .066	.475 ± .006

(b) $C = 0.1$

can be roughly ranked as follows: $IPCA < nPCA < CCIPCA < IOCA \approx EOCA < SGA \approx GHA$. Compared with most methods, EOCA is, indeed, a fast method, with small computational complexity and actual runtime.

VI. CONCLUSION

In this article, we propose a novel subspace orthogonal basis updating algorithm. Based on the principal angles between two given subspaces whose dimensionalities may be different, the proposed basis updating algorithm is able to merge them into one. Therefore, it can be employed in online incremental subspace learning. Combining the online orthonormal component extraction and updating, we design an incremental algorithm EOCA for dimensionality reduction. As a universal unsupervised algorithm, EOCA enjoys the advantages of low computational complexity and automatic target dimensionality estimation. The experimental results demonstrate that EOCA is efficient and effective with more stability on noisy data sets.

TABLE VI
COMPARISON OF CC AND ACTUAL RUNTIME OF DIFFERENT METHODS

Theoretical	IPCA	CCIPCA	SGA	GHA	nPCA	IOCA	EOCA
CC(all)	$O(Ndk^2)$	$O(C_1Ndk)$	$O(C_2Ndk)$	$O(C_3Ndk)$	$O(C_4N(dk^2+kO(g(t)))$	$O(C_5Ndk)$	$O(C_6Ndk)$
Runtime(s)	IPCA	CCIPCA	SGA	GHA	nPCA	IOCA	EOCA
Hill	4.09	0.42	0.24	0.25	2.12	0.02	0.02
IJCNN1	4.8	2.9	0.40	0.39	3.1	0.32	0.34
Ionosphere	0.05	0.03	<0.01	<0.01	0.02	<0.01	0.01
ISOLET	96.5	23.0	12.0	12.0	35.9	6.8	2.5
Musk	1.76	0.39	0.15	0.15	1.57	0.02	0.02
MNIST	319.8	140.3	81.2	79.3	178.8	80.3	83.4
Optdigits	2.50	0.72	0.07	0.07	1.68	0.07	0.08
protein	111.4	48.5	17.0	16.7	87.8	9.99	7.90
Sonar	0.05	0.02	<0.01	<0.01	0.04	<0.01	<0.01
SensIT	54.3	15.3	2.2	2.2	6.6	1.87	2.06
Waveform	0.11	0.08	0.01	0.01	0.07	0.02	0.02
USPS	50.09	5.16	2.16	2.29	6.20	0.84	0.95
cifar10	5938.0	2812.5	433.5	419.2	3554.9	618.7	943.3

In the future, the EOCA method can be improved in the following aspect. EOCA is proposed as a linear dimensionality reduction algorithm, which may not fit nonlinearly embedded data. One solution is to combine the power of kernel technique with EOCA.

REFERENCES

[1] D. Pandove, S. Goel, and R. Rani, "Systematic review of clustering high-dimensional and large datasets," *ACM Trans. Knowl. Discovery from Data*, vol. 12, no. 2, pp. 16:1–16:68, 2018.

[2] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *J. Mach. Learn. Res.*, vol. 16, pp. 2859–2900, 2015.

[3] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *London, Edinburgh, Dublin Phil. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, Nov. 1901.

[4] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 27, no. 5, pp. 684–698, 2005.

[5] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.

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

[7] W. Sun *et al.*, "UL-isomap based nonlinear dimensionality reduction for hyperspectral imagery classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 89, pp. 25–36, Mar. 2014.

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

[9] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured AutoEncoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.

[10] X. Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, and J. T. Zhou, "Deep clustering with sample-assignment invariance prior," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 31, 2020, doi: 10.1109/TNNLS.2019.2958324.

[11] W. Sun *et al.*, "Nonlinear dimensionality reduction via the ENH-LTSA method for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observat. Remote Sens.*, vol. 7, no. 2, pp. 375–388, Feb. 2014.

[12] W. Sun, G. Yang, B. Du, L. Zhang, and L. Zhang, "A sparse and low-rank near-isometric linear embedding method for feature extraction in hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 4032–4046, Jul. 2017.

[13] K. Yuan, I. Chatzinikolaidis, and Z. Li, "Bayesian optimization for whole-body control of high-degree-of-freedom robots through reduction of dimensionality," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2268–2275, Jul. 2019.

[14] O. Elnaggar and R. Arelhi, "A new unsupervised short-utterance based speaker identification approach with parametric t-SNE dimensionality reduction," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Okinawa, Japan, Feb. 2019, pp. 92–101.

[15] K. Kim, "An improved semi-supervised dimensionality reduction using feature weighting: Application to sentiment analysis," *Expert Syst. Appl.*, vol. 109, pp. 49–65, Nov. 2018.

[16] 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.

[17] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *CoRR*, vol. abs/1704.02532, Apr. 2017.

[18] P. Hall, D. Marshall, and R. Manin, "Incremental eigenalysis for classification," in *Proc. Brit. Mach. Vis. Conf.*, vol. 1, Sep. 1998, pp. 286–295.

[19] P. Hall, D. Marshall, and R. Martin, "Merging and splitting eigenspace models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 9, pp. 1042–1049, Sep. 2000.

[20] J. Weng, Y. Zhang, and W.-S. Hwang, "Candid covariance-free incremental principal component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 1034–1040, Aug. 2003.

[21] J. Ye, R. Janardan, and Q. Li, "GPCA: An efficient dimension reduction scheme for image compression and retrieval," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Seattle, WA, USA, Aug. 2004, pp. 354–363.

[22] C.-X. Ren and D.-Q. Dai, "Incremental learning of bidirectional principal components for face recognition," *Pattern Recognit.*, vol. 43, no. 1, pp. 318–330, Jan. 2010.

[23] F. Xu, G. Gu, X. Kong, P. Wang, and K. Ren, "Object tracking based on two-dimensional PCA," *Opt. Rev.*, vol. 23, no. 2, pp. 231–243, Apr. 2016.

[24] S. Pang, S. Ozawa, and N. Kasabov, "Incremental linear discriminant analysis for classification of data streams," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 35, no. 5, pp. 905–914, Oct. 2005.

[25] J. Ye, Q. Li, H. Xiong, H. Park, R. Janardan, and V. Kumar, "IDR/QR: An incremental dimension reduction algorithm via QR decomposition," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 9, pp. 1208–1222, Sep. 2005.

[26] H. Zhao and P. C. Yuen, "Incremental linear discriminant analysis for face recognition," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 1, pp. 210–221, Feb. 2008.

[27] T.-K. Kim, B. Stenger, J. Kittler, and R. Cipolla, "Incremental linear discriminant analysis using sufficient spanning sets and its applications," *Int. J. Comput. Vis.*, vol. 91, no. 2, pp. 216–232, Jan. 2011.

[28] K. Díaz-Chito, F. J. Ferri, and W. D. Villanueva, "Image recognition through incremental discriminative common vectors," in *Proc. 12th Int. Conf. Adv. Concepts Intell. Vis. Syst. (ACIVS)*, Sydney, NSW, Australia, Dec. 2010, pp. 304–311.

[29] G.-F. Lu, J. Zou, and Y. Wang, "Incremental learning of discriminant common vectors for feature extraction," *Appl. Math. Comput.*, vol. 218, no. 22, pp. 11269–11278, Jul. 2012.

[30] D. Wang and H. Lu, "On-line learning parts-based representation via incremental orthogonal projective non-negative matrix factorization," *Signal Process.*, vol. 93, no. 6, pp. 1608–1623, Jun. 2013.

[31] K. Díaz-Chito, F. J. Ferri, and A. Hernández-Sabaté, "An overview of incremental feature extraction methods based on linear subspaces," *Knowl.-Based Syst.*, vol. 145, pp. 219–235, Apr. 2018.

[32] Y. Li, "On incremental and robust subspace learning," *Pattern Recognit.*, vol. 37, no. 7, pp. 1509–1518, Jul. 2004.

[33] F. Liu, M. Liu, T. Zhou, Y. Qiao, and J. Yang, "Incremental robust nonnegative matrix factorization for object tracking," in *Proc. 23rd Int. Conf. Neural Inf. Process. (ICONIP)*, Kyoto, Japan, Oct. 2016, pp. 611–619.

[34] T. Zhu, Y. Xu, F. Shen, and J. Zhao, "An online incremental orthogonal component analysis method for dimensionality reduction," *Neural Netw.*, vol. 85, pp. 33–50, Jan. 2017.

[35] T. Zhu, Y. Xu, F. Shen, and J. Zhao, "Orthogonal component analysis: A fast dimensionality reduction algorithm," *Neurocomputing*, vol. 177, pp. 136–146, Feb. 2016.

[36] D. Cai and X. He, "Orthogonal locality preserving indexing," in *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Salvador, Brazil, Aug. 2005, pp. 3–10.

[37] Z. Yang and E. Oja, "Linear and nonlinear projective nonnegative matrix factorization," *IEEE Trans. Neural Netw.*, vol. 21, no. 5, pp. 734–749, May 2010.

[38] J. Ye, "Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems," *J. Mach. Learn. Res.*, vol. 6, pp. 483–502, Dec. 2005.

[39] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, nos. 3–4, pp. 321–377, Dec. 1936.

[40] L. Wang, X. Wang, and J. Feng, "Subspace distance analysis with application to adaptive Bayesian algorithm for face recognition," *Pattern Recognit.*, vol. 39, no. 3, pp. 456–464, Mar. 2006.

[41] Å. Björck, "Solving linear least squares problems by gram-Schmidt orthogonalization," *BIT Numer. Math.*, vol. 7, no. 1, pp. 1–21, Mar. 1967.

[42] M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[43] D. Prokhorov, *IJCNN 2001 Neural Network Competition*. Dearborn, MI, USA: Ford Research Laboratory, 2001.

[44] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[45] J. Y. Wang, "Application of support vector machines in bioinformatics," M.S. thesis, Dept. Comput. Sci. Inf. Eng., Nat. Taiwan Univ., Taipei, Taiwan, 2002.

[46] M. F. Duarte and Y. Hen Hu, "Vehicle classification in distributed sensor networks," *J. Parallel Distrib. Comput.*, vol. 64, no. 7, pp. 826–838, Jul. 2004.

[47] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994.

[48] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.

[49] M. Artac, M. Jogan, and A. Leonardis, "Incremental PCA for on-line visual learning and recognition," in *Proc. Object Recognit. Supported Interact. Service Robots*, vol. 3, Aug. 2002, pp. 781–784.

[50] I. Dagher and R. Nachar, "Face recognition using IPCA-ICA algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 6, pp. 996–1000, Jun. 2006.

[51] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Netw.*, vol. 5, no. 6, pp. 927–935, Nov. 1992.

[52] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Netw.*, vol. 2, no. 6, pp. 459–473, Jan. 1989.

[53] J. Karhunen and J. Joutsensalo, "Representation and separation of signals using nonlinear PCA type learning," *Neural Netw.*, vol. 7, no. 1, pp. 113–127, Jan. 1994.

Tianyue Zhang received the B.Sc. degree from Nanjing University, Nanjing, China, in 2017, where she is currently pursuing the Ph.D. degree in computer science.

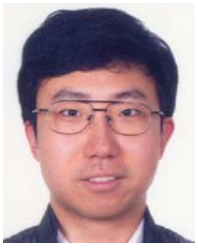
Her current research interests include online learning and incremental learning.

Furao Shen (Member, IEEE) received the B.Sc. and M.Sc. degrees in mathematics from Nanjing University, Nanjing, China, in 1995 and 1998, respectively, and the Ph.D. degree from the Tokyo Institute of Technology, Tokyo, Japan, in 2006.

He is currently a Full Professor of computer science and technology with Nanjing University. His current research interests include neural computing and robotic intelligence.

Tao Zhu received the M.Sc. and Ph.D. degrees from Nanjing University, Nanjing, China, in 2012 and 2018, respectively, where he is currently pursuing the Ph.D. degree.

His research interests include dimensionality reduction, face recognition, and neural computing.



Jian Zhao (Senior Member, IEEE) received the B.S. degree from Nanjing University, Nanjing, China, the M.Sc. degree from the Hamburg University of Technology, Hamburg, Germany, and the Dr. Sc. degree in electrical engineering from the Swiss Federal Institute of Technology (ETH) Zurich, Switzerland.

From 2010 to 2015, he was a Research Scientist with the Institute for Infocomm Research, A*STAR, Singapore. Currently, he is an Associate Professor with the School of Electronic Science and Engineering, Nanjing University. His research interests include deep neural networks, mathematical optimization, and wireless communication networks.

Dr. Zhao was honored with the Dengfeng Scholars Program of Nanjing University in 2015, IEEE Globecom 2008 Best Paper Award, and the 2009 Chinese Government Award for Outstanding Self-Financed Students Abroad.