Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 429451, 21 pages
http://dx.doi.org/10.1155/2014/429451

Research Article

EVD Dualdating Based Online Subspace Learning

Bo Jin,[1] Zhongliang Jing,[1] and Haitao Zhao[2]

[1] *School of Aeronautics and Astronautics, Shanghai Jiaotong University, 800 Dongchuan Road, Shanghai 200240, China*
[2] *School of Information Science and Engineering, East China University of Science and Technology, 130 Meilong Road, Shanghai 200237, China*

Correspondence should be addressed to Zhongliang Jing; zljing@sjtu.edu.cn and Haitao Zhao; ht.zhao.ecust@gmail.com

Received 10 April 2014; Accepted 25 June 2014; Published 24 July 2014

Academic Editor: Yan Liang

Conventional incremental PCA methods usually only discuss the situation of adding samples. In this paper, we consider two different cases: deleting samples and simultaneously adding and deleting samples. To avoid the NP-hard problem of downdating SVD without right singular vectors and specific position information, we choose to use EVD instead of SVD, which is used by most IPCA methods. First, we propose an EVD updating and downdating algorithm, called EVD dualdating, which permits simultaneous arbitrary adding and deleting operation, via transforming the EVD of the covariance matrix into a SVD updating problem plus an EVD of a small autocorrelation matrix. A comprehensive analysis is delivered to express the essence, expansibility, and computation complexity of EVD dualdating. A mathematical theorem proves that if the whole data matrix satisfies the low-rank-plus-shift structure, EVD dualdating is an optimal rank-k estimator under the sequential environment. A selection method based on eigenvalues is presented to determine the optimal rank k of the subspace. Then, we propose three incremental/decremental PCA methods: EVDD-IPCA, EVDD-DPCA, and EVDD-IDPCA, which are adaptive to the varying mean. Finally, plenty of comparative experiments demonstrate that EVDD-based methods outperform conventional incremental/decremental PCA methods in both efficiency and accuracy.

1. Introduction

Principal component analysis (PCA) [1], known as the subspace learning, or the Karhunen-Loeve transform [2], has been an active topic in machine learning and pattern recognition societies in the last several decades. As a well-known unsupervised linear dimension reduction and multivariate analysis method, PCA has been applied to biometric recognition [3], gene classification [4], latent semantic indexing [5], and visual tracking [6].

In order to obtain the optimal set of normal orthogonal basis, which endues PCA with the minimal reconstruction error, the batch-mode PCA can be achieved in two ways: the eigenvalue decomposition (EVD) of the data covariance matrix and the singular value decomposition (SVD) of the data matrix. Both approaches have a high computational cost and a mass demand of storage, in the case of a high-dimensional and large-scale dataset. In practical applications, not all the observations are available before training. Especially in online usage, samples arise sequentially along with time. In these situations, the batch-mode PCA does not satisfy the demand for real-time process due to its requirement to recompute the EVD or SVD of the whole data every time.

To solve this issue, incremental learning has been investigated for more than two decades in both applied mathematics and machine learning community, whose task is to update the learning results without reexecuting the whole process when adding new data points. Various effective incremental PCA (IPCA) methods have been proposed.

In a period of knowledge explosion, the fast growing information is usually adulterated with mock, invalid, or expired data. The presence of a few deviated samples might tremendously contaminate the solved model, such as principal directions in PCA. The overdue instances, which can be regarded as outliers compared to unexpired instances in some degree, could reduce the accuracy of data model. Therefore, for an intelligent learning system, the only function to

admit new instances is not enough, but the capability to eliminate aberrant samples is also necessary. This is the aim of decremental learning. Comparing with IPCA, decremental PCA (DPCA) did not receive adequate attention in the literature. Only a few methods have been proposed in the last ten years. Besides, there is no incremental decremental algorithm of subspace learning to the best of our knowledge. Similar works are only about support vector machine (SVM) [7]. Cauwenberghs and Poggio [8] propose an incremental decremental method of adding and/or deleting a single sample, and Karasuyama and Takeuchi [9] expand it to the case of multiple instances.

Because the essence of PCA is SVD or EVD in the mathematical form, the task of incremental PCA and decremental PCA is equivalent to updating and downdating SVD or EVD. In existing methods, most IPCA approaches adopt similar strategies via updating SVD. However, these tactics based on SVD may be impossible to be implemented for decremental PCA. Lorenzelli and Yao [10] point out that SVD downdating is NP-hard without knowing right singular vectors. Hall et al. [11] argue that right singular vectors of the remained matrix cannot be computed without visiting elements of the right singular vectors of the original matrix. Besides, in many practical applications, such as subspace learning [12, 13], image reconstruction [14], face recognition [15], and visual tracking [16], only left singular vectors are needed as the projection matrix, so right singular vectors are usually not stored to save memory. If the data matrix and the right singular vectors are not preserved, the position information of deleted points in the queue may be unknowable in the decremental case, which causes right singular vectors to be incomputable. The problem of incomplete position information does not arise in increment PCA, because it is a common sense that new instances are appended to the tail of queue.

Based on the demand on incremental decremental learning and the difficulty of decremental learning in the analysis above, we introduce a novel online subspace method for simultaneous incremental decremental learning. The contributions in this paper are as follows.

(1) To avoid the problem of lacking right singular vectors in decremental learning, we utilize EVD instead of SVD and propose a dualdating algorithm for eigenspace, that is, EVD dualdating, which can accept and delete samples at the same time. Our algorithm transforms the EVD updating and downdating of the covariance matrix into a SVD updating problem plus an EVD of a small autocorrelation matrix. To the best of our knowledge, it is the first attempt of simultaneous incremental decremental subspace learning and has a simpler and unitized mathematical form, which theoretically guarantees a better performance than the conventional multiple-step implementation.

(2) Several theoretical and computational analyses are presented to further explore the property of EVD dualdating, including the essence and geometric explanation of EVD dualdating, expansive forms of EVD dualdating for data revising and weighted updating, the computation complex of EVD dualdating, a mathematical theorem which demonstrates the optimality of EVD dualdating in the sequential mode if the data matrix satisfies *low-rank-plus-shift* structure, and a selection method of the optimal rank k based on eigenvalues.

(3) It is proofed that the change of mean caused by adding or deleting samples in the varying-mean PCA can be transformed into adding and deleting several equivalent vectors in the zero-mean PCA. Thus, three online PCA algorithms are derived based on EVD dualdating to cope with changeable mean: incremental PCA (EVDD-IPCA), decremental PCA (EVDD-DPCA), and incremental decremental PCA (EVDD-IDPCA).

The remaining of this paper is organized as follows. Section 2 briefly reviews the updating and downdating methods of both SVD and EVD and incremental PCA. The proposed EVD downdating algorithm and its analyses are presented in Section 3. In Section 4, EVD dualdating is applied to incremental decremental PCA with mean updating. Section 5 presents the experiment results and comparisons with other approaches. Section 6 concludes this paper. In the end, proofs of lemmas and theorems are in the Appendix.

2. Related Work

Over the past few decades, many efficient incremental PCA methods have been proposed. Generally, existing IPCA algorithms can be divided into three categories. The first category updates eigenvectors without any matrix decomposition. The typical method is the candid covariance-free IPCA (CCIPCA) [17]. The second category updates principal components via EVD updating. The subspace merging and splitting model, developed by Hall et al. [12], belong to this category. With the help of partition R-SVD [14] and SVD updating [18], the third category is the most studied which recomputes singular values and singular vectors via sequentially updating SVD.

Weng et al. [17] propose an incremental PCA method without computing the covariance matrix, the candid covariance-free IPCA(CCIPCA). The CCIPCA algorithm computes principal components sequentially and considers the complementary space of lower order PCs when calculating higher order PCs. Because the computation of the $(i+1)$th PC depends on the ith one, the error will be accumulated in the whole process. Besides, the sample-to-dimension ratio needs to be large enough to avoid some problems coming from the view of statistical estimation. This condition is not satisfied in many situations.

Hall et al. [12] develop a merging and splitting eigenspace model (MSES). This algorithm is an online subspace learning algorithm based on EVD, via solving a small eigenproblem on a new orthonormal basis. MSES is able to update or downdate EVD by adding or subtracting the eigenspace of added or deleted samples and adaptive to the change of data mean.

Except these two approaches above, other incremental PCA methods are based primarily on SVD. Levy and Lindenbaum [19] propose the sequential Karhunen-Loeve (SKL) method based on the partitioned R-SVD algorithm, which simplifies the SVD of a large data matrix into the SVD of some small ones via a sequential procedure. Then, this sequentialized partitioned R-SVD algorithm is utilized to extract PCs from a sequence of human face images. Besides, a forgetting factor is employed to weaken the affection of old data. However, in SKL, the mean of data is not taken into consideration, so the result is not accurate enough when confronting an image sequence with a variational mean, such as a human face under a changeable illumination. Skočaj and Leonadris [13] develop the weighted and robust incremental subspace learning (WR-ISL) algorithm, which has the ability to deal with the change of mean and weighted data. However, WR-ISL does not consider the chunk updating mode; in other words, only one sample can be manipulated in each round. The mean updating for multiple samples is solved by Ross et al. [16], who demonstrate that the covariance matrix of the combined data is equal to the sum of covariance matrices of the old data, the new data, and an additional vector when taking the mean into account. According to this, Ross et al. obtain an extended SKL algorithm with mean updating, which is applied to visual tracking and successfully locates one human face and one toy with different poses under variational background and illumination, both indoor and outdoor.

Zha and Simon [18] propose a more generalized mathematical formula to update SVD, namely, SVD updating. This algorithm, which is applied to LSI, is an efficient incremental method to recalculate the rank-k SVD for updating documents, updating terms, and term weight corrections. Moreover, Zha and Simon prove that if the united data matrix satisfies the *low-rank-plus-shift* structure, the result of the SVD updating algorithm with the new data and the optimal rank-k approximate of old data is still an optimal rank-k estimation. Zhao et al. [20] propose a chunk incremental PCA approach via the SVD updating algorithm, known as SVDU-IPCA. Comparing to other incremental PCA methods, SVDU-IPCA computes the eigendecomposition of the autocorrelation matrix instead of the covariance matrix. The motivation is that usually the sample number is much smaller than the data dimension in practical applications, so the dimension of the autocorrelation matrix is also smaller than the covariance matrix. Then, Zhao et al. find a strategy to update the eigendecomposition of a autocorrelation matrix by SVD updating. However, the change of mean is not considered in SVDU-IPCA, so it is not suitable for the situation with changeable mean. Besides, it suffers from the problem of growing demand for storage and computation, because the size of autocorrelation matrix is dilating along with the new data, and an additional process is needed to transfer the resulting right singular vectors and kept whole data to principal components. Huang et al. [15] propose an improved SVDU-IPCA method to handle changeable mean data and decrease the storage, where only a small package of concentrated data is saved to calculate left singular vectors.

Although a great deal of research has been accomplished about incremental subspace learning, the research on decremental learning is still inadequate in the literature. The merging and splitting eigenspace model developed by Hall et al. [12] can downdate EVD to recompute PCs when deleting some samples from the old data. Meanwhile, they claim that it is impossible to achieve SVD downdating in a closed form with their model. Brand [21] proposes a fast modification model of rank-k singular value decomposition (MSVD). As an extension of the term weight corrections form of SVD updating, MSVD is able to recompute the rank-k SVD of the modified data matrix after updating, downdating, revising, and recentering terms. However, this method does not take the mean into consideration, so its result is not accurate when the data mean was time-varying. Melenchón and Martínez [22] develop a method for downdating, composing, and splitting SVD (DCSSVD) with a changeable mean. DCSSVD accomplishes these by downdating, composing, and splitting the right singular vectors firstly, then computing the mean and SVD of the remained right singular vectors, and finally calculating the resulting SVD. However, this method suffers from a severe efficiency problem, since the core process is the SVD of a $d \times k$ matrix, whose computation complexity is $O(dk \min(d, k))$, still depending on the data dimension. AIPCA, proposed by Wang et al. [23], is a decremental version of SVDU-IPCA algorithm which recomputes the eigendecomposition of the autocorrelation matrix by MSVD. Although AIPCA achieves decremental subspace learning, it inherits disadvantages of SVDU-IPCA and MSVD, such as incapability of handling changeable mean, a large memory to preserve the data matrix, and an additional process to transfer its results to left singular vectors.

Beside the accuracy and efficiency, the severest problem faced by SVD-based decremental methods is that it is a NP-hard problem without the position information of deleted samples in the data matrix, which might be not obtainable in many practical applications.

3. EVD Dualdating

In Section 3.1, we first briefly review the SVD updating [18] algorithm. Our EVD dualdating algorithm is proposed in Section 3.2. Then, the related analyses are reported in the following sections.

3.1. SVD Updating. Given a data matrix $A = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d\times n}$, its SVD is $A = U\Sigma V^T$, where $U \in \mathbb{R}^{d\times d}$, $\Sigma \in \mathbb{R}^{d\times n}$, and $V \in \mathbb{R}^{n\times n}$. The best rank-$k$ approximation of A is

$$\widehat{A} = U_k \Sigma_k V_k^T, \tag{1}$$

where Σ_k is the diagonal matrix with the largest k singular values and U_k and V_k are the first k columns of U and V, respectively. We call (1) the rank-k singular value decomposition (rank-k SVD) of A.

When new samples $B = [x_{n+1}, x_{n+2}, \ldots, x_{n+m}] \in \mathbb{R}^{d\times m}$ come, how to compute the rank-k SVD of the new data matrix $A^\bullet = [A \;\; B]$ by only using U_k, Σ_k, V_k, and B?

Input: The rank-k SVD of old data A, U_k, Σ_k, V_k, new data B.
Output: The rank-k SVD of total data $A^{\bullet}$, $\check{U}_k$, $\check{\Sigma}_k$, $\check{V}_k$.
(1) Compute QR decomposition, $(I_d - U_k U_k^T)B = \mathrm{QR}$;
(2) Compute the rank-k SVD, $\begin{bmatrix} \Sigma_k & U_k^T B \\ 0 & R \end{bmatrix} = \widetilde{U}_k \widetilde{\Sigma}_k \widetilde{V}_k^T$;
(3) The rank-k SVD of $[A \;\; B]$ is

$$\widehat{A^{\bullet}} = \left([U_k \;\; Q] \, \widetilde{U}_k \right) \widetilde{\Sigma}_k \left(\begin{bmatrix} V_k & 0 \\ 0 & I_m \end{bmatrix} \widetilde{V}_k \right)^T.$$

Algorithm 1: SVD updating.

To solve this problem, Zha and Simon [18] propose an efficient mathematical tool, namely, SVD updating. Its detailed procedure is described in Algorithm 1.

3.2. EVD Dualdating. In this section, a thorough discussion of the proposed dualdating algorithm for EVD is presented. Dualdating means updating and downdating together; in other words, we consider the situation of adding and deleting samples simultaneously.

Given a data matrix $A = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d\times n}$, its SVD is $A = U\Sigma V$. Let $S_A = AA^T \in \mathbb{R}^{d\times d}$ the covariance matrix (in this section, we do not distinguish the covariance matrix and the scatter matrix, since their difference is only the coefficient $1/n$), then S_A is a symmetric positive semidefinite matrix, and its EVD is $S_A = U\Lambda U^T$, where $\Lambda = \Sigma^2$. The best rank-k approximation of S_A is

$$\widehat{S}_A = U_k \Lambda_k U_k^T, \tag{2}$$

where U_k is the first k columns of U and Λ_k is a diagonal matrix with the largest k eigenvalues in Λ. For any matrix $A \in \mathbb{R}^{d\times n}$, we call (2) the rank-$k$ eigenvalue decomposition (rank-k EVD) of A or S_A.

Now some old samples D are to be deleted, where D can be composed of arbitrary p ($p < n$) columns in A. Without losing any generality, let D be the last p columns: $D = [x_{n+1-p}, \ldots, x_n] \in \mathbb{R}^{d\times p}$. Meanwhile, new instances B are available: $B = [x_{n+1}, \ldots, x_{n+m}] \in \mathbb{R}^{d\times m}$. We are interested in how to express the rank-k EVD of the final data matrix $A^{\star} = [x_1, \ldots, x_{n-p}, x_{n+1}, \ldots, x_{n+m}] \in \mathbb{R}^{d\times(n-p+m)}$ as modifications to U_k, Λ_k via D and B.

3.2.1. Basic Procedure. The basic procedure of the proposed EVD dualdating algorithm is as follows. Let

$$\Phi = \begin{bmatrix} I_n & & \\ & I_m & \\ & & -I_p \end{bmatrix} \in \mathbb{R}^{(n+m+p)\times(n+m+p)}. \tag{3}$$

Thus, the covariance matrix of $A^{\star}$ can be written as

$$\begin{aligned} A^{\star} A^{\star T} &= AA^T + BB^T - DD^T \\ &= [A \;\; B \;\; D] \, \Phi [A \;\; B \;\; D]^T. \end{aligned} \tag{4}$$

The basic idea of EVD dualdating is to transform the dualdating problem into a SVD updating problem plus an extra process with a small computation complexity. Firstly, consider the matrix $[A \;\; B \;\; D]$. Knowing U_k, $\Sigma_k = \Lambda_k^{1/2}$, we assume that right singular vectors in the rank-k SVD of A are V_k. Then, the rank-k SVD of $[A \;\; B \;\; D]$ can be calculated by the SVD updating algorithm:

$$\widehat{[A \;\; B \;\; D]} = \check{U}_k \check{\Sigma}_k \check{V}_k^T, \tag{5}$$

where $\check{U}_k \in \mathbb{R}^{d\times k}$, $\check{\Sigma}_k \in \mathbb{R}^{k\times k}$, and $\check{V}_k \in \mathbb{R}^{(n+m+p)\times k}$.

Take (5) into (4); we have

$$A^{\star} A^{\star T} = \check{U}_k \check{\Sigma}_k \check{V}_k^T \Phi \check{V}_k \check{\Sigma}_k^T \check{U}_k^T. \tag{6}$$

Let

$$\Psi = \check{\Sigma}_k \check{V}_k^T \Phi \check{V}_k \check{\Sigma}_k^T \in \mathbb{R}^{k\times k}. \tag{7}$$

Because $A^{\star} A^{\star T}$ is a symmetric positive semidefinite matrix, Ψ is also symmetric positive semidefinite. Usually $k \ll d$, so Ψ is a small matrix compared to the covariance matrix of $A^{\star}$. The EVD of Ψ is

$$\Psi = P \Lambda^{\star} P^T, \tag{8}$$

where $P \in \mathbb{R}^{k\times k}$, $\Lambda^{\star} \in \mathbb{R}^{k\times k}$ is the diagonal eigenvalue matrix.

Finally, the rank-k EVD of $A^{\star} A^{\star T}$ is

$$A^{\star} A^{\star T} = \left(\check{U}_k P \right) \Lambda^{\star} \left(\check{U}_k P \right)^T. \tag{9}$$

By (3) to (9), we have successfully converted the dualdating problem of EVD into a SVD updating problem of adding $m + p$ samples plus an EVD of a small $k \times k$ matrix.

3.2.2. Further Simplification. Although the basic procedure of our EVD dualdating algorithm is given, one problem still remains unsolved: the assumed right singular vectors V_k is unobtainable. Here we address this problem by simplifying the computation of Ψ.

Consider the results of the SVD updating algorithm on the rank-k SVD of $[A \;\; B \;\; D]$:

$$\begin{aligned} \check{U}_k &= [U_k \;\; Q] \, \widetilde{U}_k, \qquad \check{\Sigma}_k = \widetilde{\Sigma}_k, \\ \check{V}_k &= \begin{bmatrix} V_k & & \\ & I_m & \\ & & I_p \end{bmatrix} \widetilde{V}_k, \end{aligned} \tag{10}$$

Input: The rank-k EVD of old data $A \in \mathbb{R}^{d\times n}$, U_k, Λ_k, the deleted data $D \in \mathbb{R}^{d\times p}$, the added data $B \in \mathbb{R}^{d\times m}$.
Output: The rank-k EVD of the remained data $A^\star \in \mathbb{R}^{d\times(n+m-p)}$, $U_k^\star$, $\Lambda_k^\star$.

(1) Let equivalent adding data matrix $B^\star = \begin{bmatrix} B & D \end{bmatrix}$.

(2) Compute QR decomposition, $(I_d - U_k U_k^T)B^\star = \mathrm{QR}$.

(3) $\Sigma_k = \Lambda_k^{1/2}$, compute the rank-$k$ SVD, $\begin{bmatrix} \Sigma_k & U_k^T B^\star \\ 0 & R \end{bmatrix} = \widetilde{U}_k \widetilde{\Sigma}_k \widetilde{V}_k^T$.

(4) Let $\Phi = \begin{bmatrix} I_k & & \\ & I_m & \\ & & -I_p \end{bmatrix}$, calculate $\Psi = \widetilde{\Sigma}_k \widetilde{V}_k^T \Phi \widetilde{V}_k \widetilde{\Sigma}_k$.

(5) Compute the EVD of Ψ, $\Psi = P\Lambda^\star P^T$.

(6) The rank-k EVD of $A^\star$ is

$$U_k^\star = \left(\begin{bmatrix} U_k & Q \end{bmatrix} \widetilde{U}_k\right) P, \qquad \Lambda_k^\star = \Lambda^\star.$$

Algorithm 2: EVD dualdating.

where $U_k \in \mathbb{R}^{d\times k}$, $Q \in \mathbb{R}^{d\times(m+p)}$, $\widetilde{U}_k \in \mathbb{R}^{(k+m+p)\times k}$, $\check{U}_k \in \mathbb{R}^{d\times k}$, $\widetilde{\Sigma}_k \in \mathbb{R}^{k\times k}$, $V_k \in \mathbb{R}^{n\times k}$, $\widetilde{V}_k \in \mathbb{R}^{(k+m+p)\times k}$, and $\check{V}_k \in \mathbb{R}^{(n+m+p)\times k}$.

Take (10) into (7); we have

$$\begin{aligned} \Psi &= \widetilde{\Sigma}_k \widetilde{V}_k^T \begin{bmatrix} V_k^T V_k & & \\ & I_m & \\ & & -I_p \end{bmatrix} \widetilde{V}_k \widetilde{\Sigma}_k^T \\ &= \widetilde{\Sigma}_k \widetilde{V}_k^T \begin{bmatrix} I_k & & \\ & I_m & \\ & & -I_p \end{bmatrix} \widetilde{V}_k \widetilde{\Sigma}_k^T. \end{aligned} \tag{11}$$

From (11), it can be seen that the right singular vectors V_k are actually not needed, and the computation of Ψ is simplified.

3.2.3. Algorithm. The detailed procedure of EVD dualdating has been presented above. To sum up, the pseudocode of our EVD dualdating algorithm is described in Algorithm 2. To achieve pure updating or downdating of EVD, it is only needed to let B or D be an empty set. From the computation progress, EVD dualdating is degenerated into the standard SVD updating in the updating mode.

3.3. Analysis of EVD Dualdating. In this section, we first analyze the mechanism of EVD dualdating for incremental and decremental learning. Second, some extended forms of EVD dualdating are given for particular uses. Third, the computation complexity on the proposed EVD dualdating algorithm is presented. Fourth, the optimality of EVD dualdating in the sequential usage is demonstrated. Finally, we discuss how to determine the optimal rank k.

3.3.1. Mechanism of Incremental and Decremental Learning. For convenience, when analysing the essence of incremental and decremental learning based on EVD dualdating, we only consider the pure updating or downdating situation and denote the changed matrix as $B \in \mathbb{R}^{d\times m}$ in both situations.

According to the procedure of EVD dualdating, two key decompositions are the SVD updating of the equivalent adding matrix $\begin{bmatrix} A & B \end{bmatrix}$ and the EVD of the small matrix Ψ. And in the SVD updating algorithm, the core step is the rank-k SVD of the following matrix:

$$H \triangleq \begin{bmatrix} \Sigma_k & U_k^T B \\ 0 & R \end{bmatrix} = \widetilde{U}_k \widetilde{\Sigma}_k \widetilde{V}_k^T, \tag{12}$$

where Σ_k is the diagonal variance matrix of the original data A, $U_k^T B$ is the coefficient matrix by projecting B onto the subspace spanned by U_k, and R is the upper triangular reconstruction error matrix of B. So a conclusion can be obtained that, in H, the left k columns represent the original data, and the remained m columns represent the added or deleted data.

Then, divide the columns of $\widetilde{V}_k$ into two partitions:

$$\widetilde{V}_k = \begin{bmatrix} \widetilde{V}_k^1 \\ \widetilde{V}_k^2 \end{bmatrix}, \tag{13}$$

where $\widetilde{V}_k^1$ are the first k rows of $\widetilde{V}_k$ and $\widetilde{V}_k^2$ are the last m rows of $\widetilde{V}_k$. Thus, $\widetilde{V}_k^1$, $\widetilde{V}_k^2$ stand for the old data and the changed data, respectively.

So the matrix Ψ can be written as

$$\begin{aligned} \Psi &= \widetilde{\Sigma}_k \begin{bmatrix} \widetilde{V}_k^{1T} & \widetilde{V}_k^{2T} \end{bmatrix} \begin{bmatrix} I_k & \\ & \pm I_m \end{bmatrix} \begin{bmatrix} \widetilde{V}_k^1 \\ \widetilde{V}_k^2 \end{bmatrix} \widetilde{\Sigma}_k^T \\ &= \widetilde{\Sigma}_k \widetilde{V}_k^{1T} \widetilde{V}_k^1 \widetilde{\Sigma}_k^T \pm \widetilde{\Sigma}_k \widetilde{V}_k^{2T} \widetilde{V}_k^2 \widetilde{\Sigma}_k^T. \end{aligned} \tag{14}$$

Now, let us observe the situation from the view of geometry shown in Figure 1. On the left is the column space of data. The red arrows represent the orthogonal basis U_k of the old subspace. The green arrow is the added or deleted samples B, whose projection on U_k and reconstruction error are the green dashed arrows. After QR decomposition, a new basis in the extended subspace is made up by the red and pink arrows. However, the projection of the data matrix on this basis is not completely diagonalized. So, the SVD of the coefficient matrix H on this basis is executed to obtain the diagonalizing matrix. Then, the new orthogonal basis after adding B is represented by the blue arrows. At this time,

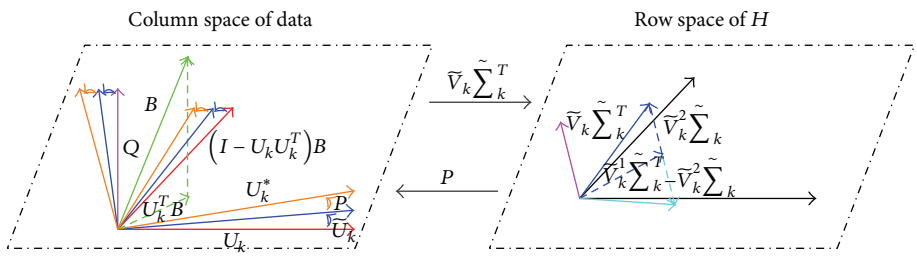

Figure 1: Visualization of EVD dualdating.

Table 1: Expansions of EVD dualdating.

	Old data	New data	Φ	Miscellanea
Update	$A_{d\times n}$	$\begin{bmatrix} A_{d\times n} & B_{d\times m} \end{bmatrix}$	$\begin{bmatrix} I_k & \\ & I_m \end{bmatrix}$	
Downdate	$\begin{bmatrix} C_{d\times(n-p)} & D_{d\times p} \end{bmatrix}$	$C_{d\times(n-p)}$	$\begin{bmatrix} I_k & \\ & -I_p \end{bmatrix}$	
Dualdate	$\begin{bmatrix} C_{d\times(n-p)} & D_{d\times p} \end{bmatrix}$	$\begin{bmatrix} C_{d\times(n-p)} & B_{d\times m} \end{bmatrix}$	$\begin{bmatrix} I_k & & \\ & I_m & \\ & & -I_p \end{bmatrix}$	
Revise	$\begin{bmatrix} C_{d\times(n-p)} & D_{d\times p} \end{bmatrix}$	$\begin{bmatrix} C_{d\times(n-p)} & B_{d\times p} \end{bmatrix}$	$\begin{bmatrix} I_k & & \\ & I_p & \\ & & -I_p \end{bmatrix}$	
Weighted update	$A_{d\times n}$	$\begin{bmatrix} w_1 A_{d\times n} & WB_{d\times m} \end{bmatrix}$	$\begin{bmatrix} w_1 I_k & \\ & W \end{bmatrix}$	$W = \mathrm{diag}(w_2, \ldots, w_{m+1})$

the row space of H is drawn on the right of the figure. The black and pink arrows compose a standard orthogonal basis, where the black ones are the elements corresponding to old samples, and the green one is the elements corresponding to changed samples. The blue arrow represents the orthogonal vectors $\widetilde{V}_k \widetilde{\Sigma}_k^T$ in the row space of H after adding B via SVD updating. Because EVD dualdating adds samples B at first no matter whether the case is incremental or decremental, so it needs to make an adjustment in the row space of H. If it is deletion, the elements corresponding to changed samples are sign-changed. As shown in Figure 1, the component, marked by the cyan blue dashed arrow, is reversed. From (14), Ψ is in fact the sum or difference of autocorrelation matrices of the old data $\widetilde{V}_k^1 \widetilde{\Sigma}_k^T$ and the changed data $\widetilde{V}_k^2 \widetilde{\Sigma}_k^T$. According to the relationship between the column and row space, the EVD of Ψ is utilized to acquire the new rotation matrix P in the data space. Finally, the resulting orthogonal basis $U_k^\star$ is the orange arrows.

To sum up, the aim of EVD dualdating is to obtain the projection matrix caused by the change of sample set, and the essence of EVD dualdating is to transform the EVD of a varying covariance matrix in the data space to the EVD of a varying autocorrelation matrix in a dimension-reduced row space.

3.3.2. Extendibility of EVD Dualdating. From the deduction of EVD dualdating, it can be seen that nearly no restriction is imposed on B, D, and Φ. In the downdating mode, the procedure is still feasible even if D is not columns of A. Meanwhile, Φ can be selected as any matrix which matches the dimension. The only condition needed to be satisfied is that Ψ must be a positive semidefinite matrix. From another view, our EVD dualdating algorithm has a favorable extendibility.

The standard dualdating mode for EVD dualdating is adding and deleting samples synchronously. As we mentioned before, when B or D is an empty set, EVD dualdating can work at the pure incremental or decremental mode. When $m = p$, EVD dualdating can be seen as data revising. Another useful extension is forgetting updating, or the so-called weighted updating, which is very important for online applications. In the learning procedure, prior instances should be assigned a low weight since they become antiquated as time goes on. Without proper weighting mechanisms, the contribution of too many old similar samples can become too prominent that new instances seem meaningless. In [16], the forgetting factor is used to destrengthen the effect of old images of the tracked object. Via EVD dualdating, a concise but meticulous weighting formula can be acquired, in which the weight of an arbitrary sample can be modulated, similar as the way adopted in [13]. The sole operation is modifying Φ as follows:

$$\Phi = \begin{bmatrix} w_1 I_k & \\ & W \end{bmatrix}, \qquad W = \begin{bmatrix} w_2 & & \\ & \ddots & \\ & & w_{m+1} \end{bmatrix}. \tag{15}$$

The equivalent data matrix of weighted updating is $\begin{bmatrix} w_1 A & WB \end{bmatrix}$. The detailed expansions of EVD dualdating are listed in Table 1.

3.3.3. Computation Complexity of EVD Dualdating. Before analyzing, we define some signs to simplify the representation: QR(m,n) and SVD(m,n) stand for the QR and SVD decomposition of a $m \times n$ rectangle matrix and QR(m), SVD(m), and EVD(m) stand for the QR, SVD, and EVD decomposition of a $m \times m$ square matrix. The computation of the proposed EVD dualdating algorithm is composed of four parts: QR of $(I_{d\times d} - U_k U_k^T)B^\star$, SVD of H, EVD of Ψ, and other multiplication operations including calculating reconstruction error, Ψ, and $U_k^\star$. Because the first k column of $(I_{d\times d} - U_k U_k^T)B^\star$ is already orthogonal, its QR decomposition only operates the last $m + p$ columns actually. The computation complexity is presented in Table 2, where the computation is presented into two parts: matrix decomposition and transformation cost.

In the pure updating or downdating mode, there are two matrix decompositions in our EVD-Dualdating algorithm, one more than other pure updating and downdating methods. This may cause EVD dualdating slower than other methods. But taking the dimension and the transformation cost into account, the efficiency of EVD dualdating is close or even better, comparing to other methods. The main advantage of our algorithm can be reflected in the dualdating mode. As the only method achieving simultaneous updating and downdating, EVD dualdating can avoid many repeating processes and decrease the cumulative error. An experimental comparison of efficiency and accuracy on our EVD dualdating and other incremental and decremental methods is presented in Section 5.

3.4. Justification of the Sequential Usage of U_k and Λ_k. In many online applications, it is impossible to store the original data because of the limitation of the physical medium and the consideration about efficiency. Described in a mathematical form, this means that the original data matrix A is unobtainable and replaced by its best rank-k approximation which can be calculated by U_k and Λ_k. So, it is urgent to demonstrate the effectiveness of EVD dualdating in a sequential process.

Zha and Simon [18] proof that when the combined matrix satisfied the *low-rank-plus-shift* structure, SVD updating is optimal when A is replaced by its best rank-k approximation. Here, a theoretical demonstration is given to illustrate that if the whole data matrix satisfies the *low-rank-plus-shift* structure, the result of EVD dualdating after any adding or deleting operations is also an optimal rank-k estimation. First, we state Lemma 1 without proof.

Lemma 1. *Let $A \in \mathbb{R}^{d\times n}$, $d > n$, and its EVD is $S_A = AA^T = U_A \Lambda_A U_A^T = \sum_{i=1}^n \lambda_i u_{Ai} u_{Ai}^T$, where u_{Ai} and λ_{Ai} are the ith eigenvector and eigenvalue, respectively. Then for $\forall p > k$, one has*

$$\text{best}_k(S_A) = \text{best}_k\left(S_A - \sum_{i=p}^{n} \lambda_{Ai} u_{Ai} u_{Ai}^T\right). \tag{16}$$

The lemma above indicates that for the rank-k EVD it is safe to cutoff the minor eigenspaces, without affecting the optimality. With this, we discuss under the *low-rank-plus-shift* structure, when A is replaced by $\text{best}_k(A)$, the information discarded will also be discarded after EVD dualdating. The conclusion is summarized in Theorem 2, whose proof can be found in the Supplementary Material (available online at http://dx.doi.org/10.1155/2014/429451).

Theorem 2. *Given a matrix $A \in \mathbb{R}^{d\times n}$, with its* best-$k$ *approximation $\widehat{A}$, the deleted data $D \in \mathbb{R}^{d\times p}$ from A, the added data $B \in \mathbb{R}^{d\times m}$, $d > n > p$. Let $C = [A \ \underline{D}]$ be the remained data from A, $F = [C \ D \ B]$ the full data, and $E = [C \ B]$ the final data, where the underline means deletion. Let $\widehat{C} = [\text{best}_k(A) \ \underline{D}]$ be the remained matrix after deleting columns corresponding to D from A's* best-k *approximation, and let $\widehat{E} = [\widehat{C} \ B]$ be the final data from $\widehat{A}$. Assume that F satisfies the low-rank-plus-shift structure; that is,*

$$F^T F = X_F + \sigma^2 I_n, \quad \sigma > 0, \tag{17}$$

where X_F is symmetric and positive semidefinite with $\text{rank}(X_F) = k$*; then*

$$\text{best}_k(S_E) = \text{best}_k(\widehat{S}_E). \tag{18}$$

3.5. Criterion for the Optimal Rank k Selection. In the deduction above, the rank of subspace is assumed to be a fixed number k; however the optimal dimension of subspace depends on a priori information, which is possibly unknown in practical applications. Based on the fact that the bulk of variability of a given dataset can be captured by the top few eigenvectors, we introduce an eigenvalue-based method to determine the best rank k of subspace during the online learning procedure.

Supposing the truncation operation is not yet executed in steps 4 and 6 of Algorithm 2, the ranks of obtained $U_k^\star$ and $\Lambda_k^\star$ are $k + m + p$. First, we define the rate r_i:

$$r_i = \frac{\sum_{j=1}^{i} \lambda_j}{\sum_{j=1}^{k+m+p} \lambda_j}, \tag{19}$$

which indicates the proportion of the first i eigenvalues in all eigenvalues. Then, the best dimension k can be selected as the minimum number i exceeding some threshold:

$$k = \min\{i \mid r_i \geq T_r\}, \tag{20}$$

where the threshold T_r is a value in $(0, 1)$. In the batch mode, the threshold only depends on the proportion of information to be preserved. For EVD dualdating, because the estimation of rank k and truncation are performed in every round, the threshold is relative to the ratio of saved and new information. In practical implementation, it can be chosen according to the chunk size of added and deleted samples.

4. Incremental Decremental PCA Based on EVD Dualdating

In the deduction of EVD dualdating in Section 3.2, the mean of samples is not considered, but in practical applications, centralization is a necessary process to reduce the effect of

Table 2: Computation complexity of EVD dualdating and other updating/downdating methods.

	Updating		Downdating	
	Decomposition	Transformation	Decomposition	Transformation
MSES	$\mathrm{SVD}(d,m)+\mathrm{QR}(d,m)+\mathrm{EVD}(k+m)$	$O(d(k+m)^2)$	$\mathrm{SVD}(d,p)+\mathrm{EVD}(k)$	$O(dk(k+m))$
DCSSVD			$\mathrm{QR}(n-p,k)+\mathrm{SVD}(d,k)$	$O((d+n)k^2)$
MSVD	$\mathrm{QR}(d,m)+\mathrm{QR}(n+m,m)+\mathrm{SVD}(k+m)$	$O((d+n+m)(k+m)^2)$	$\mathrm{QR}(d,p)+\mathrm{QR}(n,p)+\mathrm{SVD}(k+p)$	$O((d+n)(k+p)^2)$
AIPCA	$\mathrm{QR}(k+m,m)+\mathrm{QR}(n+k,m)+\mathrm{SVD}(k+m)$	$O((d+k+m)(n+m)(k+m))$	$\mathrm{QR}(k,p)+\mathrm{QR}(n,p)+\mathrm{SVD}(k+p)$	$O(dn(k+p))$
	Dualdating			
	decomposition		Transformation	
EVDD	$\mathrm{QR}(d,m+p)+\mathrm{SVD}(k+m+p)+\mathrm{EVD}(k)$		$O(dk(k+m+p))$	

Input: The sample number, mean, eigenvectors, eigenvalues of old data, n, μ_A, U_k, Λ_k, the sample number and mean of new samples B, m, μ_B.
Output: The sample number, mean, eigenvectors, eigenvalues of new data, $n^\star$, $\mu^\star$, $U_k^\star$, $\Lambda_k^\star$.
(1) Update the sample number and mean, $n^\star = n + m$, $\mu^\star = (n/(n+m))\mu_A + (m/(n+m))\mu_B$.
(2) Compute the extra added sample, $x_a = \sqrt{(n+m)/nm}(\mu_A - \mu_B)$.
(3) Equivalent added data matrix, $\widehat{B} = [B - \mu_B \mathbf{1}_m, x_a]$.
(4) Compute $U_k^\star$, $\Lambda_k^\star$, via EVD-Dualdating with U_k, Λ_k, and added data $\widehat{B}$.

Algorithm 3: EVDD-base incremental PCA (EVDD-IPCA).

Input: The sample number, mean, eigenvectors, eigenvalues of old data, n, μ_A, U_k, Λ_k, the sample number and mean of deleted samples D, p, μ_D.
Output: The sample number, mean, eigenvectors, eigenvalues of remained data, $n^\star$, $\mu^\star$, $\Lambda_k^\star$, $U_k^\star$.
(1) Update the sample number and mean, $n^\star = n - p$, $\mu^\star = (n/(n-p))\mu_A - (p/(n-p))\mu_D$.
(2) Compute the extra deleted sample, $x_d = \sqrt{np/(n-p)}(\mu_A - \mu_D)$.
(3) Equivalent deleted data matrix, $\widehat{D} = [D - \mu_D \mathbf{1}_p, x_d]$.
(4) Compute $U_k^\star$, $\Lambda_k^\star$ via EVD-Dualdating with U_k, Λ_k and deleted data $\widehat{D}$.

Algorithm 4: EVDD-base decremental PCA (EVDD-DPCA).

environment. In this section, we first provide a brief review of PCA. Then, we take mean into account and propose three online subspace learning algorithms: EVDD-IPCA, EVDD-DPCA, and EVDD-IDPCA.

4.1. Principal Component Analysis. Principal component analysis (PCA) is one of the most popular multivariate analysis and dimension reduction methods. The goal of PCA is to find a set of normal orthogonal basis, so-called principal components, which has the best reconstruction performance in the sense of minimum mean squared error (MMSE).

Given a data matrix $A = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^{d\times n}$, the covariance matrix of A is defined by $S_A = (1/n)\sum_{i=1}^{n}(x_i - \mu_A)(x_i - \mu_A)^T$. Principal components (PCs) are the first k eigenvectors $\{u_1, \ldots, u_k\}$ corresponding to the largest k eigenvalues $\{\lambda_1, \ldots, \lambda_k\}$. Let $U = \{u_1, \ldots, u_k\}$, $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_k)$; then U and Λ can be achieved by the EVD of the covariance matrix, $S_A = U\Lambda U^T$. Another way of solving PCA is to compute the SVD of the centralized data matrix $A - \mu\mathbf{1}_n = U\Sigma V^T$, where $\mathbf{1}_n$ stands for a $1 \times n$ full-1 row vector, each column of left singular vectors U is a principal component, and $\Sigma = \sqrt{n\Lambda}$ is the singular value matrix.

4.2. Incremental and Decremental PCA. When confronting a huge dataset with a high dimension, both batch-mode methods, no matter EVD or SVD, cost tremendous time and storage. Besides, for an online learning system, it has to face an awkward circumstance that not all the instances are available before training, or some expired instances need to be deleted after training. Obviously, these problems exceed the ability of the batch-mode PCA. The incremental and decremental PCA is a natural solution.

In this section, we consider EVD dualdating with a time-varying mean, and deduce the incremental decremental PCA formula based on EVD dualdating. As mentioned before, EVD dualdating degenerates into SVD updating without right singular vectors in the updating mode, so EVDD-IPCA is actually the same as the extended sequential KL algorithm. Nonetheless we still present it in this paper for integrity. The interested reader can find more details in the reference paper [16].

The key idea of EVDD-based incremental and decremental PCA algorithm is that centralizing the original samples, the added samples, and the deleted samples separately and utilizing some mean-revising vectors to keep the covariance matrix equal to the original one. The methods of determining these mean-revising vectors are introduced in Lemmas 3, 4, and 5. For incremental or decremental PCA, there is only one mean-revising vector, noted as the equivalent added vector x_a or the equivalent deleted vector x_d, respectively, which is direct ratio to the difference of the original mean and the changed sample mean. For increment decremental PCA, the situation is a little more complex. Because of the existence of intersecting items, there are three mean-revising vectors, two equivalent added vectors x_{a1}, x_{a2}, and one equivalent deleted vector x_d. Based on these lemmas, the proposed EVDD-IPCA, EVDD-DPCA, and EVDD-IDPCA algorithms are presented in Algorithms 3, 4, and 5.

4.2.1. Incremental PCA

Lemma 3. *Let $A = \{x_1, x_2, \ldots, x_n\}$, $B = \{x_{n+1}, x_{n+2}, \ldots, x_{n+m}\}$ be two data matrices, and let their concatenation be $E = [A \;\; B]$. Denote the means and scatter matrices of A, B, and E as μ_A, μ_B, and μ_E and S_A, S_B, and S_E, respectively. This holds*

$$S_E = S_A + S_B + x_a x_a^T, \tag{21}$$

where $x_a = \sqrt{((n+m)/nm)}(\mu_A - \mu_B)$.

Input: The sample number, mean, eigenvectors, eigenvalues of old data, n, μ_A, U_k, Λ_k, the sample number and mean of added samples B, m, μ_B, the sample number and mean of deleted samples D, p, μ_D.
Output: The sample number, mean, eigenvectors, eigenvalues of remained data, $n^\star$, $\mu^\star$, $U_k^\star$, $\Lambda_k^\star$.
(1) Update the sample number and mean, $n^\star = n + m - p$, $\mu^\star = (n\mu_A + m\mu_B - p\mu_D)/(n + m - p)$.
(2) Compute the extra sample, $x_{a1} = \sqrt{n}(\mu_A - \mu^\star)$, $x_{a2} = \sqrt{m}(\mu_B - \mu^\star)$, $x_d = \sqrt{p}(\mu_D - \mu^\star)$.
(3) Equivalent added data matrix, $\widehat{B} = [B - \mu_B \mathbf{1}_m, x_{a1}, x_{a2}]$.
(4) Equivalent deleted data matrix, $\widehat{D} = [D - \mu_D \mathbf{1}_p, x_d]$.
(5) Compute $U_k^\star$, $\Lambda_k^\star$ via EVD-Dualdating with U_k, Λ_k and added data $\widehat{B}$, deleted data $\widehat{D}$.

Algorithm 5: EVDD-base incremental decremental PCA (EVDD-IDPCA).

4.2.2. Decremental PCA

Lemma 4. *Let $C = \{x_1, x_2, \ldots, x_{n-p}\}$, $D = \{x_{n-p+1}, x_{n-p+2}, \ldots, x_n\}$ be two data matrices, and let $A = [C \;\; D]$ be their concatenation. Denote the means and scatter matrices of C, D, and A as μ_C, μ_D, and μ_A and S_C, S_D, and S_A, respectively. This holds*

$$S_C = S_A - S_D - x_d x_d^T, \tag{22}$$

where $x_d = \sqrt{(np/(n-p))}(\mu_A - \mu_D)$.

4.2.3. Incremental Decremental PCA

Lemma 5. *Let $C = \{x_1, x_2, \ldots, x_{n-p}\}$, $D = \{x_{n-p+1}, x_{n-p+2}, \ldots, x_n\}$, and $B = \{x_{n+1}, x_{n+2}, \ldots, x_{n+m}\}$ be three data matrices, and let $A = [C \;\; D]$, $E = [C \;\; B]$. Denote the means and scatter matrices of A, D, B, and E as μ_A, μ_D, μ_B, and μ_E and S_A, S_D, S_B, and S_E, respectively. This holds*

$$S_E = S_A + S_B + x_{a1} x_{a1}^T + x_{a2} x_{a2}^T - S_D - x_d x_d^T, \tag{23}$$

where $x_{a1} = \sqrt{n}(\mu_A - \mu_E)$, $x_{a2} = \sqrt{m}(\mu_B - \mu_E)$, and $x_d = \sqrt{p}(\mu_D - \mu_E)$.

Remark 6. As an important approach of dimension reduction, PCA is utilized as the preprocessing method for many other machine learning methods, and the feature extraction method in other applications. Because these methods usually work in the subspace of PCA, there is a great demand to achieve simultaneous online incremental decremental subspace learning and data reconstruction. Artac et al. [24] propose a method to sequentially compute the coefficients of a sample in IPCA. Here, we introduce an incremental approach to update the projection coefficients of a data point after renewing the subspace via EVD-IDPCA, without storing the original data. For any sample x_i, assuming the eigenvectors is U_k and the mean is μ when it is added into the dataset, the reconstruction of x_i is $\widehat{x}_i = U_k c_i + \mu$, where $c_i = U^T x_i$ is the projection coefficients of x_i on the basis U_k. Then, at each round of EVD-IDPCA, the projection coefficients of x_i can be updated by

$$c_i^\star = \left(\widetilde{U}_k^{(1:k)} P\right)^T c_i + U_k^{\star T}\left(\mu - \mu^\star\right), \tag{24}$$

where $\widetilde{U}_k^{(1:k)}$ is the first k rows of $\widetilde{U}_k$. It is worth noticing that in (24) $(\widetilde{U}_k^{(1:k)} P)^T$ is a procedure variable in EVD dualdating, and $U_k^{\star T}(\mu - \mu^\star)$ only needs to be computed once for all samples, so the computational amount of updating c_i is small, $O(k^2)$, but the memory to store a data point is reduced from $O(d)$ to $O(k)$.

5. Experiment

In this section, experiments of the proposed algorithms based on EVD dualdating are presented, compared with other classic methods. Because incremental PCA has been discussed a lot in the earlier literature and the proposed EVDD-IPCA is actually equivalent to the extended sequential KL algorithm, we do not verify IPCA methods in this paper any more. The interested reader can find the performance analysis and comparison in relative papers [12, 15, 16, 20]. In the following content, decremental PCA, incremental decremental PCA experiments on real-world datasets are firstly reported; then, an adaptive rank selection experiment of EVD-Dualdating on an artificial dataset is represented. All experiments are performed with Matlab, on a computer with dual-core 2.0 GHz CPU and 4 G RAM.

5.1. DPCA Experiment: Performance Evaluation on Real-World Data. In order to verify the performance and efficiency of the proposed EVDD-DPCA and EVDD-IDPCA, four datasets are used, including cases of both high dimension and huge size. The FERET [25] database is a standard dataset used for facial recognition system evaluation managed by DARPA and NIST. The AR [26] dataset is a popular face image dataset, where images are shot under different facial expressions, illumination conditions, and partial occlusions due to sun glasses and scarf. The Yale Face Database B (Yale B) [27] contains 5760 single light source images of 10 subjects each seen under 576 viewing conditions (9 poses ×64 illumination conditions). Subsets of AR, FERET, and Yalb B are employed in our simulation, which includes 952, 720, and 4050 cropped and centralized face images, respectively. The fourth database is the Columbia Object Image Library (COIL-100) [28], and it includes 7200 color images of 100 objects, which are taken at pose intervals of 5 degrees, corresponding to 72 poses per object. The detailed information of four datasets and our experiment settings are listed in Table 3.

To compare the performance of decremental learning, we implement the proposed EVDD-DPCA algorithm with the batch-mode PCA, MSES [12], MSVD [21], DCSSVD [22],

Table 3: Dataset and configuration for DPCA.

Data set	Dimension	Class	Delete	Sample	Training	Testing
FERET	92×112	120	40	6	4	2
AR	92×112	119	39	8	6	2
Yale B	25×30	90	30	45	30	15
COIL-100	25×25	100	40	72	42	30

and AIPCA [23]. First, the whole data are learned via the batch-mode PCA; then, assuming some classes are expired data, their samples are deleted chunk by chunk. In our experiment, the number of expired classes is 40 for FERET, 39 for AR, 30 for Yale B, and 40 for COIL-100, and the chunk size is 10. Every experiment is repeated 20 times to reduce the disturbance from the process scheduling of operating system and randomized grouping. The performance is mainly evaluated by their efficiency, accuracy of eigenspace, and performance of face recognition:

(i) execution time;

(ii) weighted angle between PCs of the batch-mode PCA and DPCA methods:

$$\theta_i^w = \frac{\lambda_i |\theta_i|}{\sum_{j=1}^{k} \lambda_j}, \tag{25}$$

where λ_i is the ith eigenvalue of the batch mode and θ_i is the angle between the ith PCs;

(iii) recognition rate.

5.1.1. Computational Efficiency by the Subspace Dimension k(10–200). Recalling the analysis of computation complexity in Section 3.3, the practical computational efficiency depends on the dimension of the small matrix for decomposition and the cost of transformation. From Table 2, DCSSVD has a larger computation complexity of matrix decomposition, QR$(n-p, k)$ and SVD(d, k); AIPCA has a larger computation complexity of transformation, $O(dn(k+p))$. For MSES, MSVD and our EVDD, they have close computation complexities: MSES needs one additional SVD(d, p) to extract the eigenspace model of deleted samples before subtracting; MSVD has two QR decomposition of the residual matrices in both the row and column space, QR(d, p) and QR(n, p), as well as a larger transformation cost $O((d+n)(k+p)^2)$; EVDD has one additional EVD(k) to transform updating to downdating. Therefore, when the data dimension d is high or the size of dataset n is huge, that is, $d, n \gg k, p$, DCSSVD and AIPCA achieve lower efficiencies, and MSES, MSVD, our EVDD achieve close higher efficiencies. This conclusion is also demonstrated by Figures 2(a), 2(b), 2(c), and 2(d), which show the execution time by kept PCs (k: 10–200) of MSES, DCSSVD, MSVD, AIPCA, and EVDD-DPCA, on FERET, AR, Yale B, COIL-100. From these figures, we observe that our proposed EVDD-DPCA achieves a better or comparable efficiency.

5.1.2. PC Estimation Equality to Ground-Truth PCs. In order to evaluate the accuracy, the angles between the resulting PCs of DPCA methods and the batch-mode PCA can be adopted. But, we choose the weighted angles by their corresponding eigenvalue, which are more suitable for evaluation because they emphasize the importance of the leading PCs. Figures 3(a), 3(b), 3(c), and 3(d) show the weighted angles of the first 50 PCs of DPCA methods on four datasets, when the number of kept PCs is 100, and the chunk size is 10. Figures 4(a), 4(b), 4(c), and 4(d) show the weighted angles error of the first 50 PCs of DPCA methods on different datasets by the number of kept PCs (k: 10–200), when the chunk size is 10.

From these figures, our proposed EVDD-DPCA algorithm performs the best accuracy of the eigenvector estimation. The accuracy of principal direction depends on the estimation of mean and the cut-off error. The error of mean will cause a bias of the origin for data centralizing, which may cause the direction of the resulting basis totally different in the worst situation. The cut-off error accumulates in the sequential process, so the more times the truncation happens, the lower accuracy the final result remains. The method to update the mean is the same in MSES and EVDD-DPCA, whose estimate is equal to the true mean. In DCSSVD, the new mean is updated via the right singular vectors V. However, V is cut off to the reduced dimension, so its estimation of mean is not accurate. But the inaccuracy of mean will not affect its computation of singular vectors, because the mean correction item is stripped off from V, and no data centralizing process is executed. So errors of the singular vectors in EVDD-DPCA, MSES and DCSSVD mainly come from the cut-off error. Before splitting, MSES calculates the EVD of the deleted data, whose result is cut off to the kept dimension. The step will bring in more cut-off error. DCSSVD directly deals with the right singular vectors V to achieve downdating, so it actually ignores the information of deleted samples reflected by high order PCs. In AIPCA and MSVD, the mean is not updated, so all the remained samples centralized with the old mean. Therefore, their results deviate far away from true PCs. In Figures 3(a), 3(b), 3(c), and 3(d), it can be seen that the weighted angle of our proposed EVDD-DPCA is much smaller than other methods, because of the accurate estimate of mean and the smaller cut-off error. MSES and MSVD have close performances, and AIPCA and MSVD have larger errors. The same conclusion can be obtained in Figures 4(a), 4(b), 4(c), and 4(d). The fluctuation at the beginning of these curves is because the dimension of observed PCs is increasing from 10 to 50.

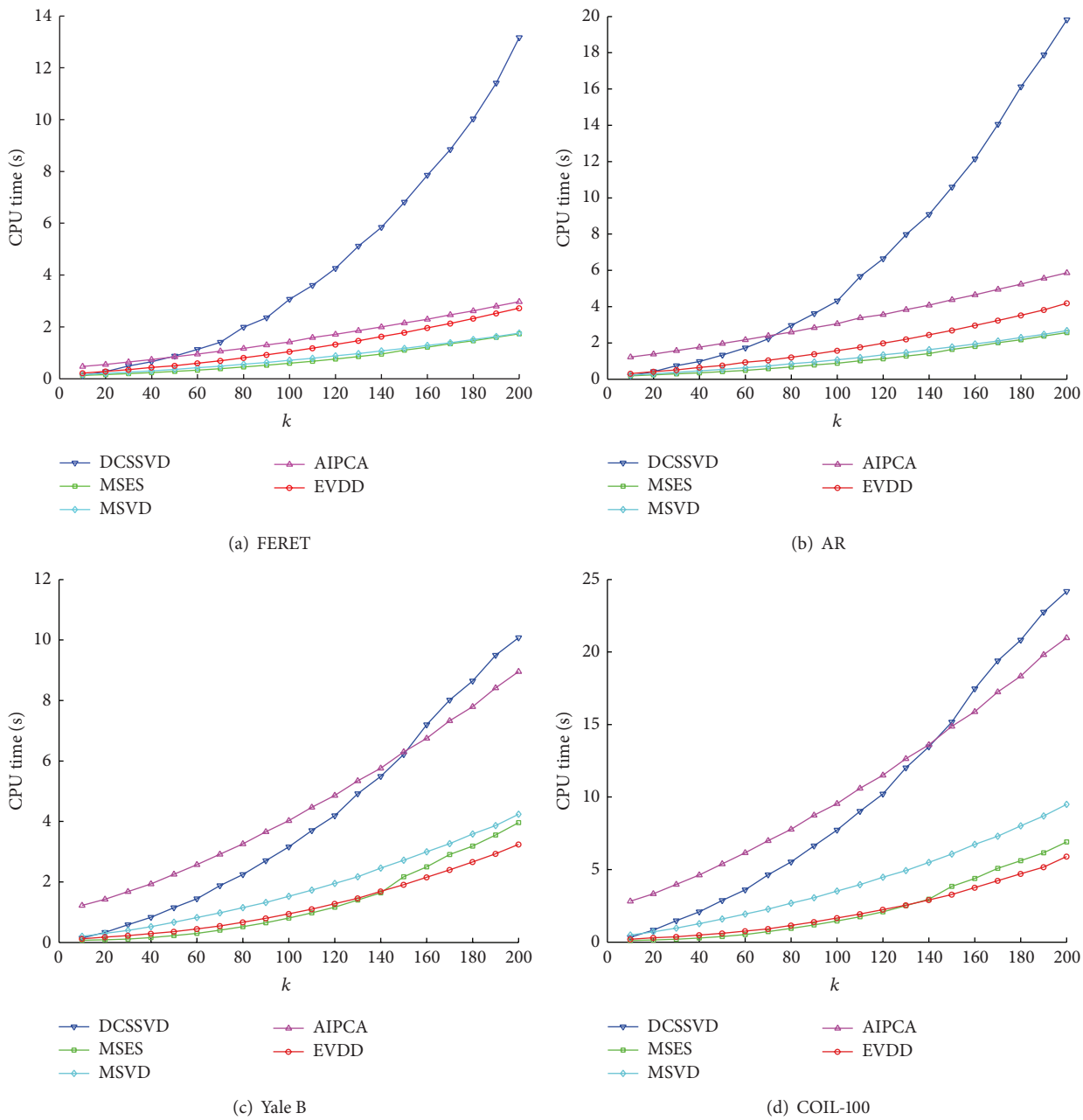

Figure 2: CPU time of DPCA methods on different datasets, k: 10–200, chunk = 10.

5.1.3. Results of Recognition with Minimum Distance Classifier. In the recognition experiment, the resulting PCs are used as the projection matrix to project the testing image to the subspace, then minimum distance classifier (MDC) is utilized for recognition. The advantage of MDC in our online application is that only the mean of each class in the projection subspace needs to be saved. The distance between a sample x and a class Ω in MDC is defined by a Mahalanobis distance:

$$d(x,\Omega) = \left[(\tilde{x} - \mu_\Omega)^T \Lambda_k^{-1} (\tilde{x} - \mu_\Omega)\right]^{1/2}, \tag{26}$$

where $\tilde{x}$ is the projection vector in the subspace, μ_Ω is the mean of the class Ω in the subspace, Λ_k is the eigenvalue matrix estimated by EVD dualdating.

Figures 5(a), 5(b), 5(c), and 5(d) represent recognition rates of the full-data PCA, the batch-mode PCA, DPCA methods. The result shows that the full-data PCA has a lower recognition rate due to the existence of expired instances, and all DPCA methods have close recognition rates, nearly equal to the batch-mode PCA. The similar results are also obtained by Ozawa et al. [29]. This phenomenon can be explained via random projection (RP) [30]. According to Johnson-Lindenstrauss lemma [31], arbitrary set of N points in a high-dimensional Euclidean space can be mapped onto a $k \geq$

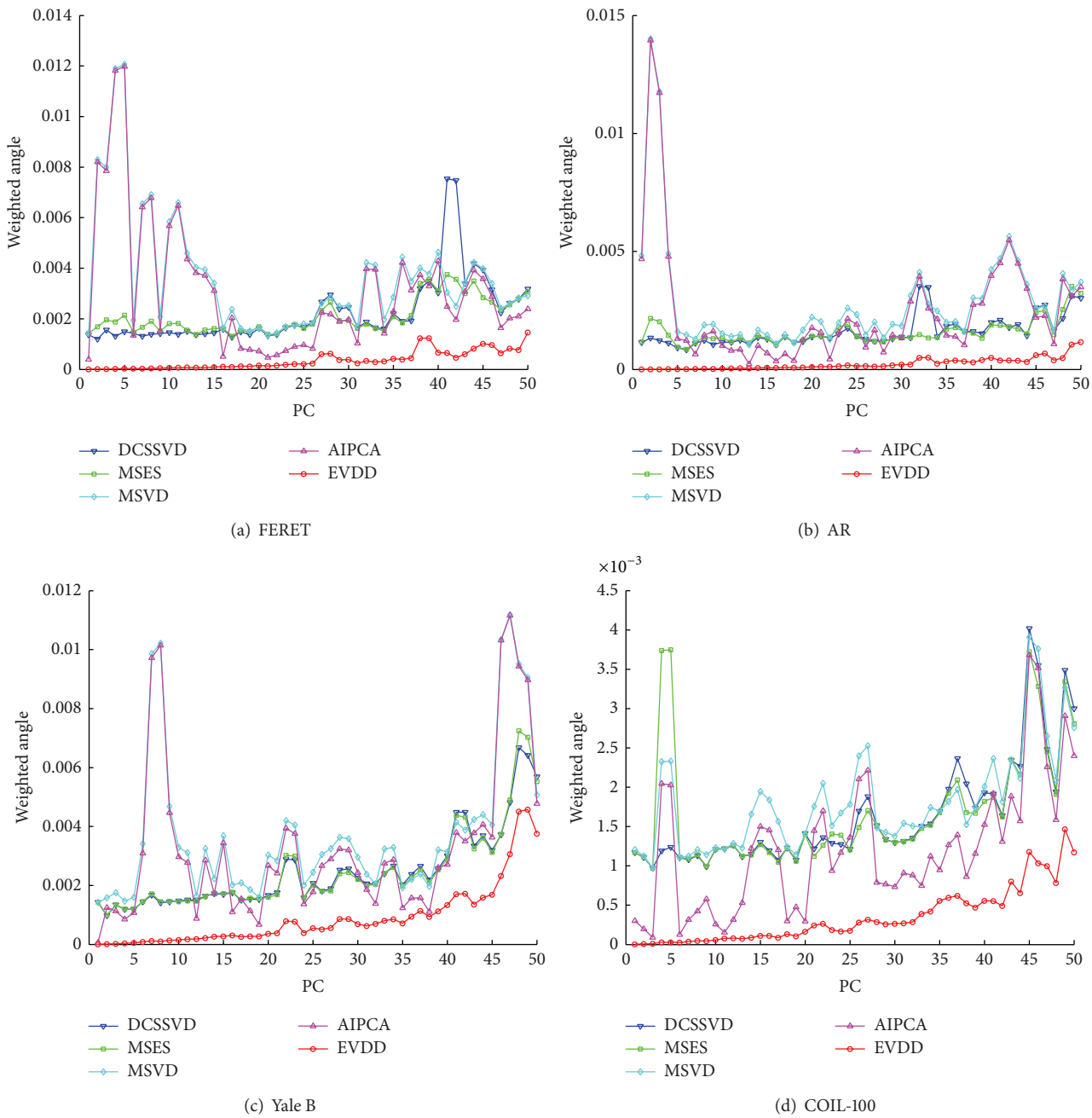

Figure 3: The first 50 weighted angles of DPCA methods on different datasets, $k = 100$, chunk = 10.

$O(\log N/\epsilon^2)$ $(0 < \epsilon < 1)$ dimension subspace where the distances between all pair of points are approximately preserved. So as long as k is large enough, for arbitrary k-dimensional random projection, the classification performance is mainly determined by MDC and the structure of data space itself. In our experiments, the smallest k is between the range [40, 60] on FERET, AR, Yale B, and about 100 on COIL-100.

5.2. IDPCA Experiment: Performance Evaluation on Real-World Data. To compare the performance of incremental decremental subspace learning methods, we implement the proposed EVDD-IDPCA algorithm with the batch-mode PCA, MSES [12], MSVD [21], DCSSVD [22], and AIPCA [23]. Because DCSSVD only accomplishes decremental PCA, we combine it with the extended SKL to achieve IDPCA. As a decremental version of SVDU-IPCA, AIPCA is connected with SVDU-IPCA to fulfill IDPCA in our experiment.

The datasets for IDPCA is the same as in the DPCA experiment and the configuration is shown in Table 4. In our experiment, samples of pretraining classes are learned by the batch-mode PCA, then at every round, a chunk of samples in expired classes are deleted, and meanwhile a chunk of samples in new classes are added. The chunk size is 10. The training/testing rate is the same as in the DPCA experiment.

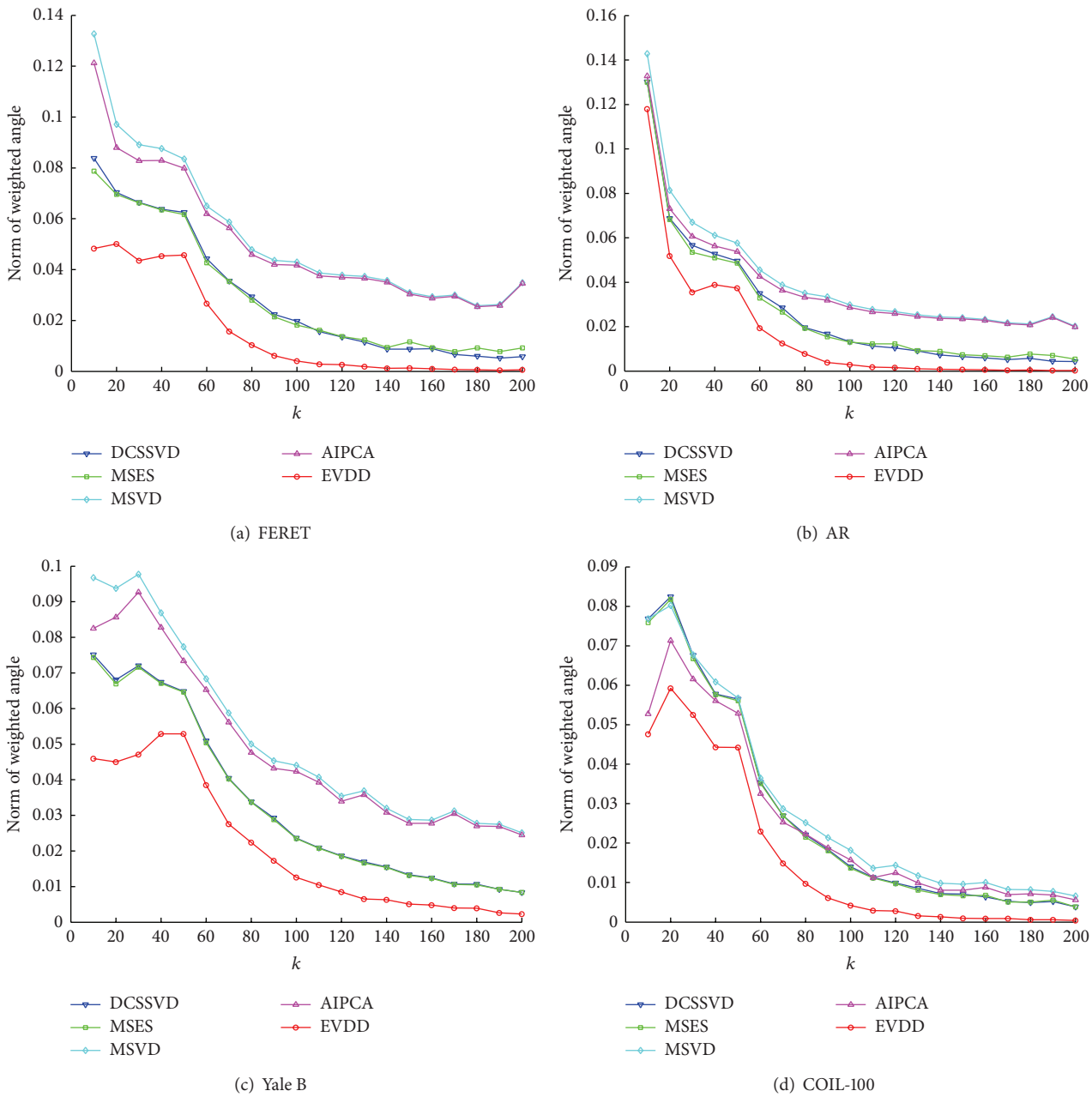

Figure 4: Error of the first 50 weighted angles of DPCA methods on different datasets, k: 10–200, chunk = 10.

Table 4: Dataset and configuration for IDPCA.

Data set	Dimension	Class	Delete	Add	Sample	Training	Testing
FERET	92×112	120	40	40	6	4	2
AR	92×112	119	39	39	8	6	2
Yale B	25×30	90	20	20	45	30	15
COIL-100	25×25	100	30	30	72	42	30

Execution time, weighted angle, and recognition rate are used to evaluate the performance of IDPCA methods.

5.2.1. Computational Efficiency by the Subspace Dimension k(10–200). Figures 6(a), 6(b), 6(c), and 6(d) present the runtime by the number of kept PCs (k) of IDPCA methods. Different from other IDPCA methods, which process incremental learning and decremental learning separately, our EVDD-IDPCA deals with deleted and added samples simultaneously, and avoids the repeating execution of

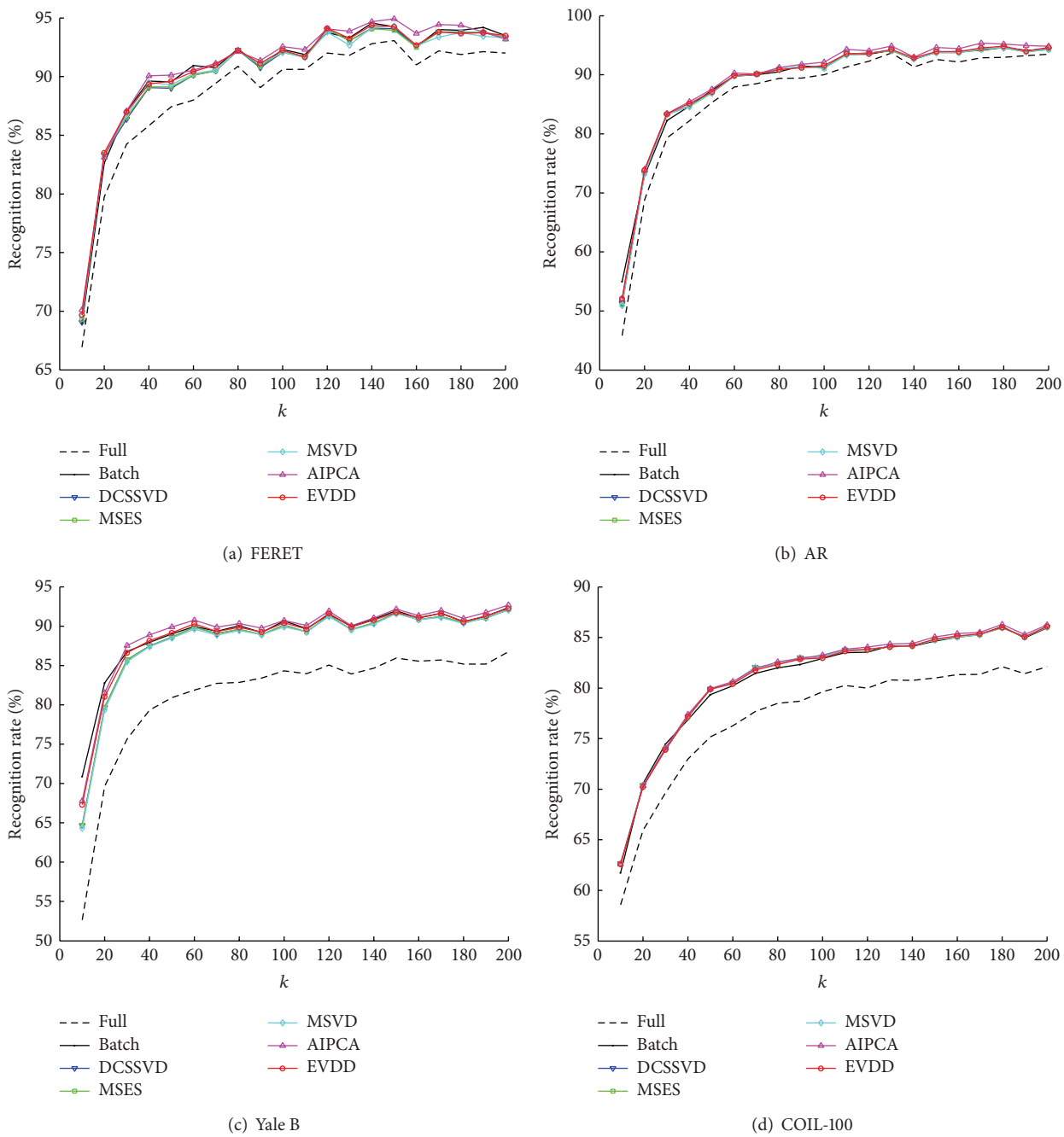

Figure 5: Recognition rate of DPCA methods on different datasets, k: 10–200, chunk = 10.

pre-processing, post-processing and some matrix decompositions. Therefore, as shown in Table 2, via the dualdating scheme, EVD dualdating has a more concise form with less matrix decompositions and a lower transformation cost. So in our experiment, the proposed EVDD-IDPCA performs much higher efficiency than other methods, especially, when the scale of dataset is large.

5.2.2. PC Estimation Equality to Ground-Truth PCs. Figures 7(a), 7(b), 7(c), and 7(d) show the weighted angles between the first 50 PCs of different IDPCA methods, when the number of kept PCs is 10, and the chunk size is 10. Figures 8(a), 8(b), 8(c), and 8(d) show the error norm of weighted angles between the first 50 PCs of IDPCA methods on different datasets by the number of kept PCs(k: 10–200), when the chunk size is 10. As the only real incremental decremental PCA method with an accurate mean estimation and a dualdating scheme, EVDD-IDPCA can obtain principal eigenvectors with fairly better approximation than other methods via avoiding redundant cut-off error. These figures show that the estimation of leading PCs in EVDD-IDPCA is significantly superior to opponents.

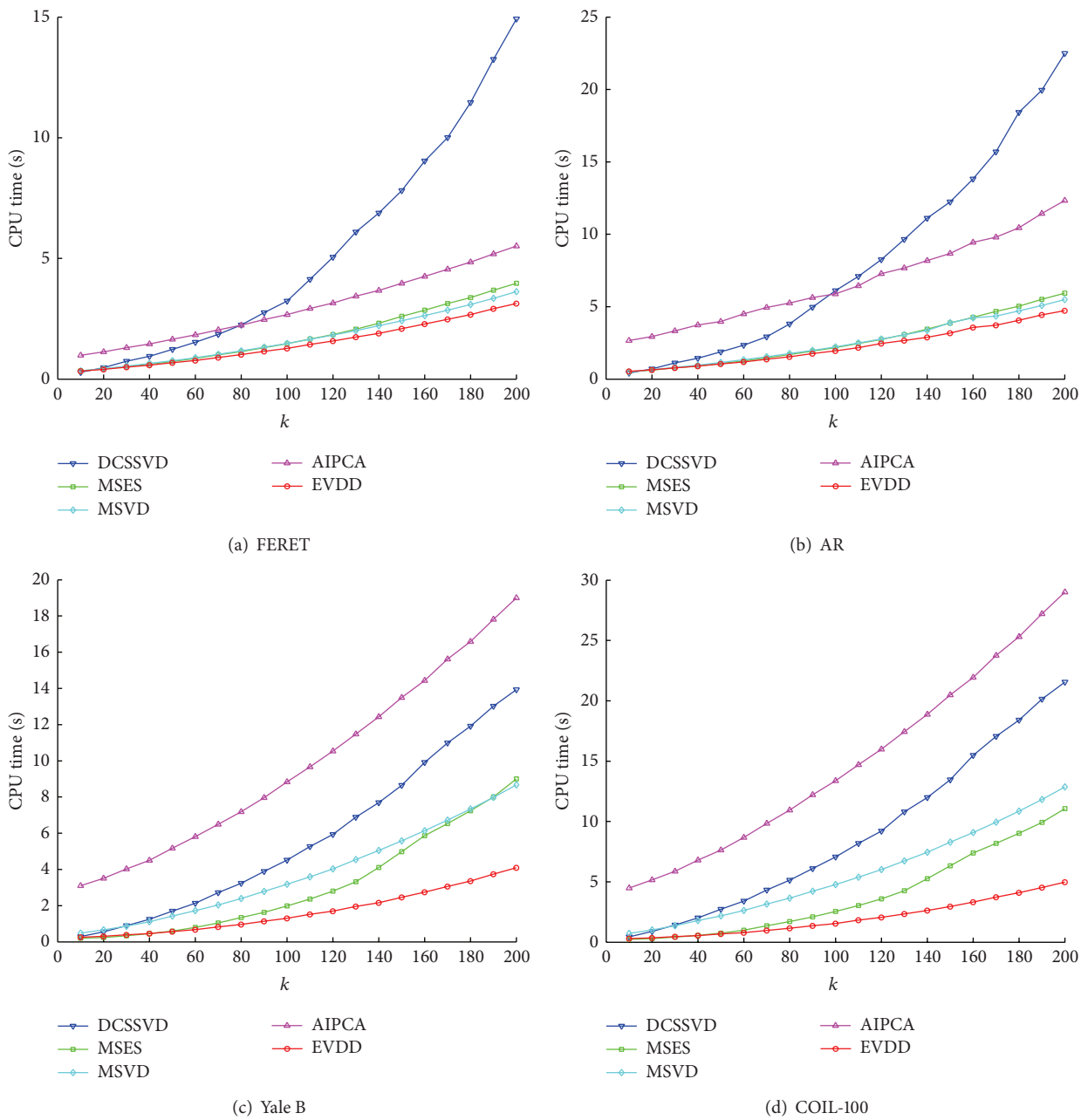

Figure 6: CPU time of IDPCA methods on different datasets, k: 10–200, chunk = 10.

5.2.3. Results of Recognition with Minimum Distance Classifier. Figures 9(a), 9(b), 9(c), and 9(d) represent recognition rates of the full-data PCA, the batch-mode PCA, IDPCA methods. The result is similar as shown in DPCA experiments, that the recognition rate of full-data PCA is lower because of the existence of expired classes, and recognition rates of considered IDPCA methods are close, mainly depending on MDC and the structure of data space, when k is large enough to satisfy Johnson-Lindenstrauss lemma.

Besides, one important advantage of EVDD-DPCA, not reflected by these DPCA and IDPCA experiments, is that the specific position information of deleted and added samples is not needed, which are necessary for DCSSVD, AIPCA, and MSVD.

5.3. Automatic Rank k Selection and Weighted EVD Dualdating. In this experiment, the selection of the dimension k of subspace without any a priori is evaluated. An artificial dataset is used here, which includes data points generated from the following model:

$$x = Ac + n, \tag{27}$$

where $A \in \mathbb{R}^{d\times d}$, $c \in \mathbb{R}^{d\times 1}$ is a coefficient vector and n is a small noise, sampled from a normal distribution $\mathcal{N}(0, \sigma I_d)$. In the simulation, the data dimension is $d = 100$, and the number of generated samples is 10000. Then, samples are sequentially learned at different chunk sizes (5, 10, 20) by our EVD dualdating and weighted EVD dualdating algorithms. The weights are $w_1 = 0.95$, and $W = I$

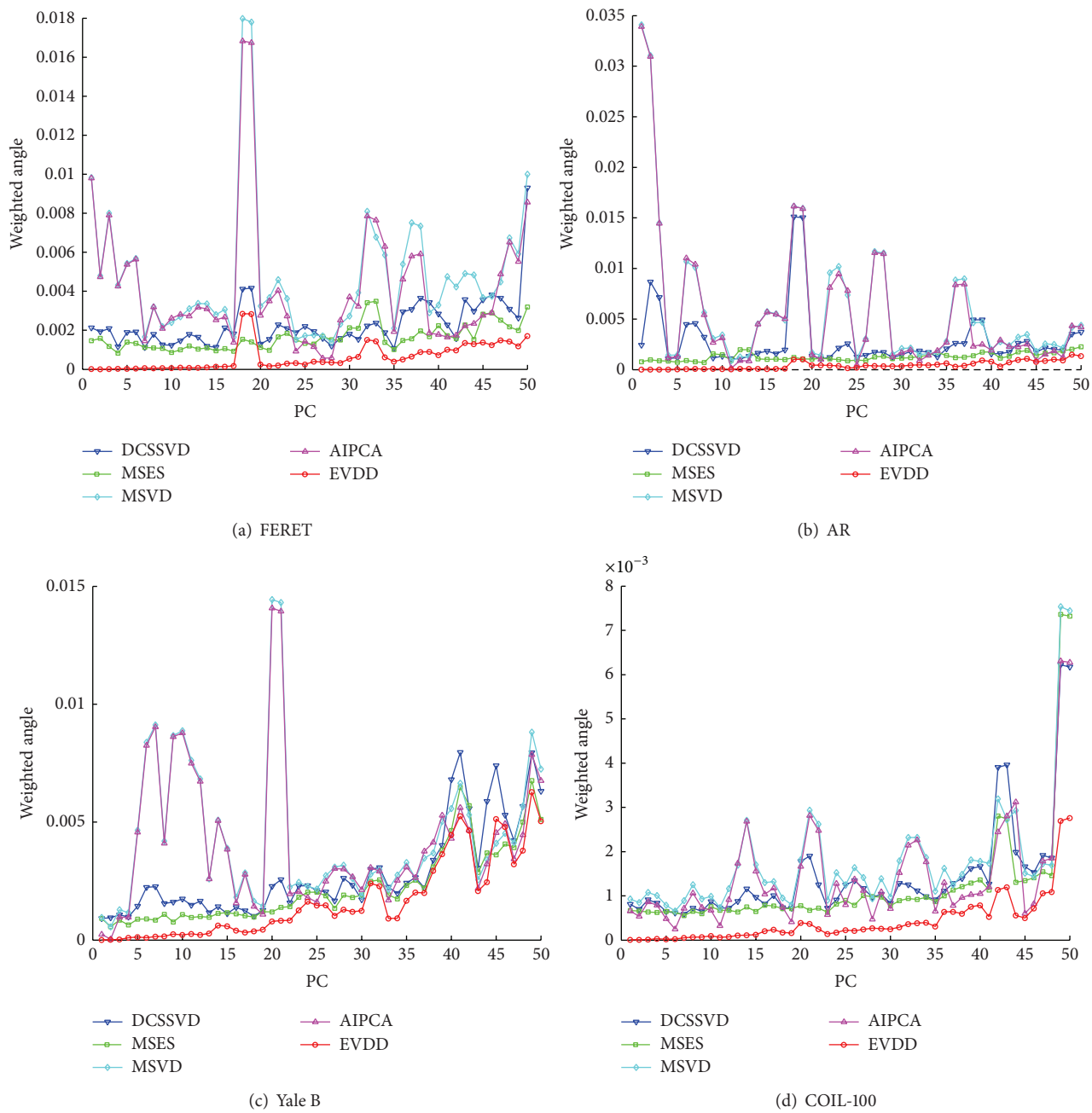

Figure 7: The first 50 weighted angles of IDPCA methods on different datasets, $k = 100$, chunk = 10.

in weighted EVD dualdating. In every round, the number of kept PCs is determined by (20), and the thresholds of preserved proportion are $T_r = 0.98, 0.97, 0.96$ with respect to the chunk sizes 5, 10, and 20. Figure 10 shows the updating curves of kept rank k during the online learning process, where the solid lines stand for weighted EVD dualdating, and the dash lines stand for EVD dualdating. From this figure, kept ranks in all curves quickly rise from the chunk size to 50–60 at the beginning, which means new features have been added to the eigenspace. Then, ranks of the weighted EVD dualdating tend to a common stable value 53. It is worth noted that the red solid line with the smallest chunk size 5 has the fast convergence speed, and the blue solid one with the largest chunk size 20 converges slowest. For normal EVD dualdating, because the influence of leading PCs is not weakened, as the online learning progresses, it becomes unwelcome to new features, and later exclusive to minor PCs. Therefore, their kept ranks, reflected by dash lines, all have a quick decreasing trend. For example, the blue dash line (chunk = 20) ends with a rank less than 30, after all samples are learned.

6. Conclusion

This paper focuses on the problem of online incremental/decremental subspace learning and reports a novel dualdating algorithm of EVD, namely, EVD dualdating. Different

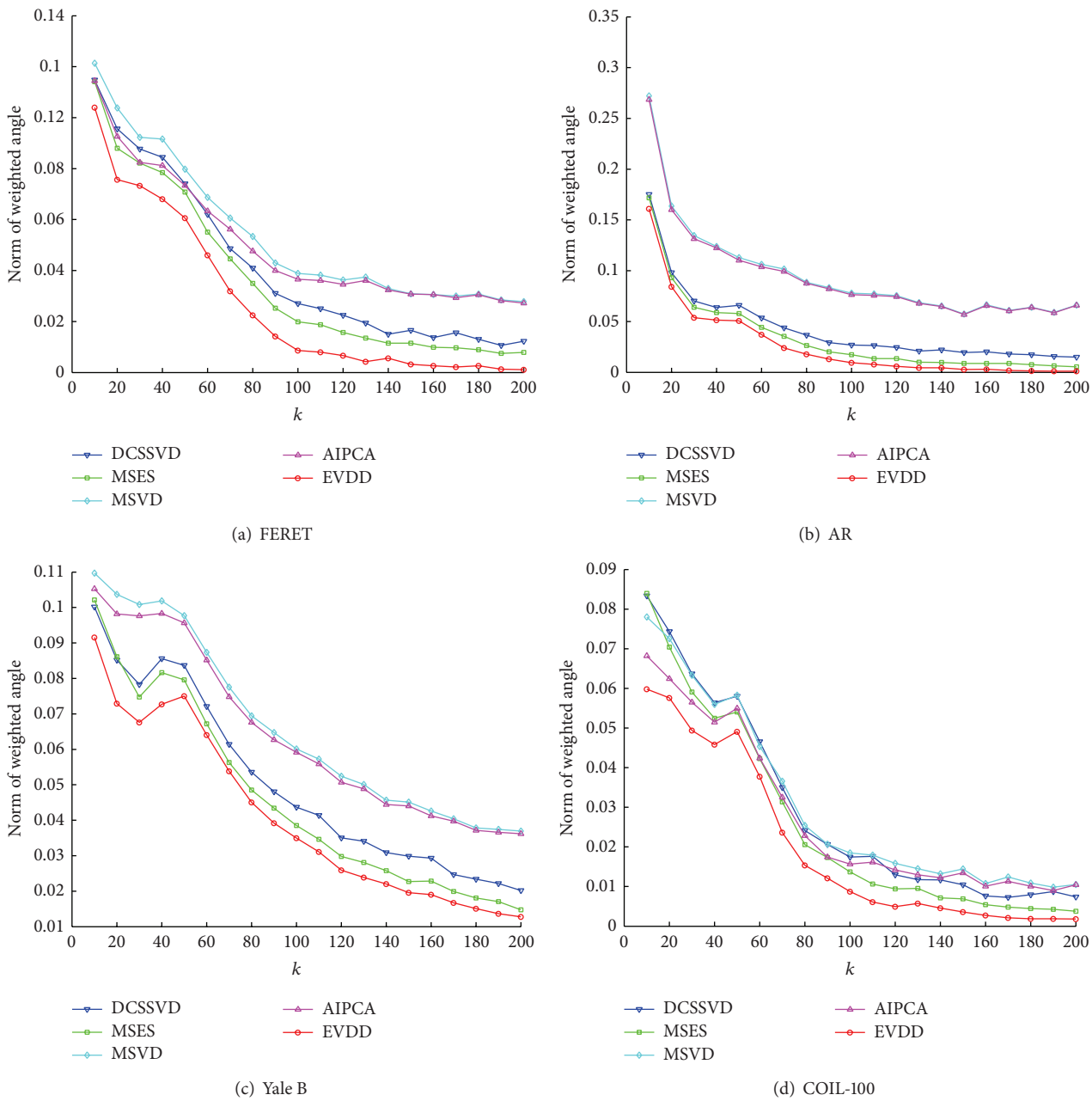

Figure 8: Error of the first 50 weighted angles of IDPCA methods on different datasets, k: 10–200, chunk = 10.

from previous works, the proposed EVD dualdating algorithm can renew the EVD of a data matrix while adding and deleting samples simultaneously. With EVD dualdating, IPCA-EVDD, DPCA-EVDD, and IDPCA-EVDD are presented to handle the changeable mean, where the variation is equivalent to add and delete several additional vectors in the case of zero-mean PCA. Plenty of comparative experiments on both real-world and artificial databases demonstrate that our EVD dualdating algorithm has a significant better approximation accuracy and computational efficiency than other state-of-the-art incremental and decremental PCA methods.

Appendices

A. Proof of Lemma 4

By definition,

$$\begin{aligned} \mu_E &= \frac{n\mu_A}{n-p} - \frac{p\mu_D}{n-p}, \\ \mu_A - \mu_E &= \frac{p}{n-p}\left(\mu_D - \mu_A\right), \\ \mu_D - \mu_E &= \frac{n}{n-p}\left(\mu_D - \mu_A\right). \end{aligned} \tag{A.1}$$

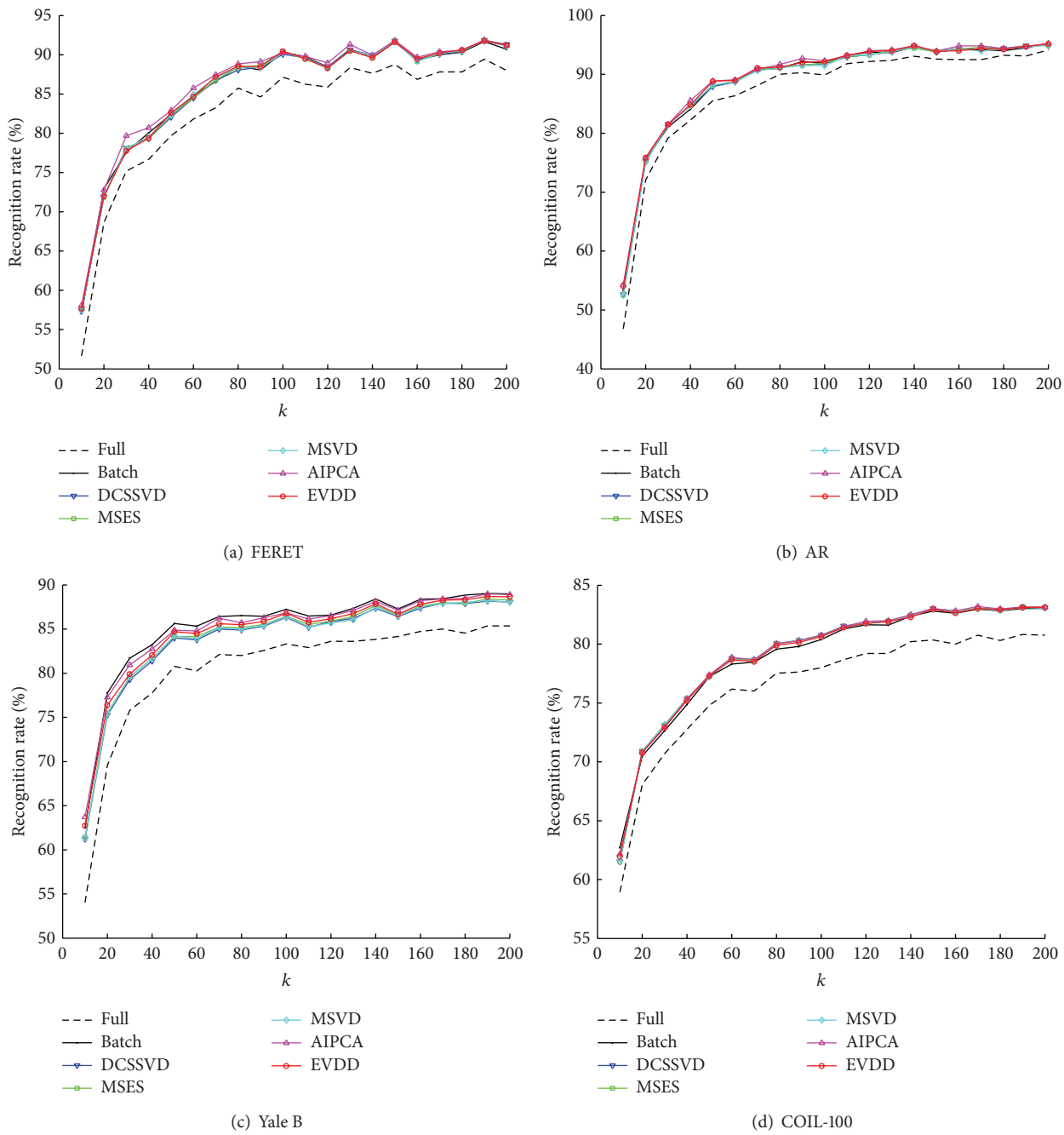

Figure 9: Recognition rate of IDPCA methods on different datasets, k: 10–200, chunk = 10.

And, the scatter matrix of E is

$$\begin{aligned} S_E &= \sum_{i=1}^{n} (x_i - \mu_E)(x_i - \mu_E)^T - \sum_{i=n-p+1}^{n} (x_i - \mu_E)(x_i - \mu_E)^T \\ &= S_A + n(\mu_A - \mu_E)(\mu_A - \mu_E)^T \\ &\quad - S_D - p(\mu_D - \mu_E)(\mu_D - \mu_E)^T \\ &= S_A - S_D + \frac{np^2}{n-p}(\mu_D - \mu_A)(\mu_D - \mu_A)^T \\ &\quad - \frac{n^2 p}{n-p}(\mu_D - \mu_A)(\mu_D - \mu_A)^T \\ &= S_A - S_D - \frac{np}{n-p}(\mu_D - \mu_A)(\mu_D - \mu_A)^T. \end{aligned} \tag{A.2}$$

B. Proof of Lemma 5

By definition,

$$\mu_E = \frac{n\mu_A + m\mu_B - p\mu_D}{n+m-p}. \tag{B.1}$$

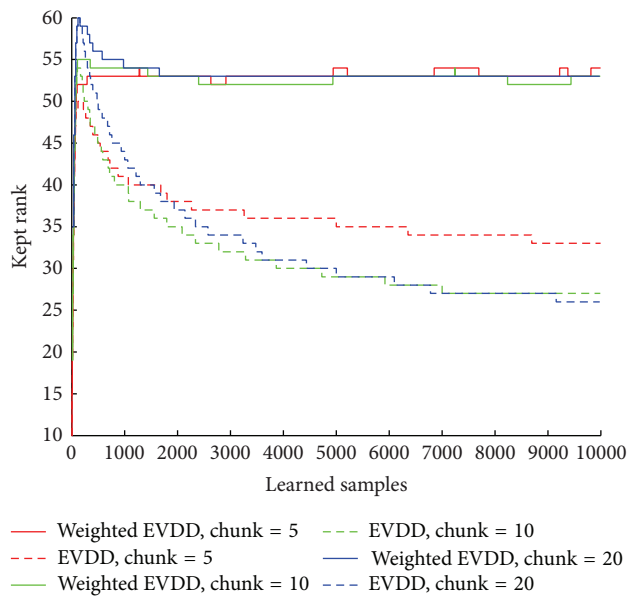

Figure 10: Results of rank k selection of EVD dualdating and weighted EVD dualdating, chunk = 5, 10, 20.

And, the scatter matrix of C is

$$\begin{aligned} S_C &= \sum_{i=1}^{n} (x_i - \mu_E)(x_i - \mu_E)^T + \sum_{i=n+1}^{n+m} (x_i - \mu_E)(x_i - \mu_E)^T \\ &\quad - \sum_{i=n-p+1}^{n} (x_i - \mu_E)(x_i - \mu_E)^T \\ &= S_A + n(\mu_A - \mu_E)(\mu_A - \mu_E)^T \\ &\quad + S_B + m(\mu_B - \mu_E)(\mu_B - \mu_E)^T \\ &\quad - S_D - p(\mu_D - \mu_E)(\mu_D - \mu_E)^T. \end{aligned} \tag{B.2}$$

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grants nos. 61175028, 61375007), the Ph.D. Programs Foundation of Ministry of Education of China (Grants nos. 20090073110045), and the Shanghai Pujiang Program (Project no. 12PJ1402200).

References

[1] I. T. Jolliffe, *Principal Component Analysis*, Springer, New York, NY, USA, 2nd edition, 2002.

[2] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 1990.

[3] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[4] K. Y. Yeung and W. L. Ruzzo, "Principal component analysis for clustering gene expression data," *Bioinformatics*, vol. 17, no. 9, pp. 763–774, 2001.

[5] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[6] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. van den Hengel, "A survey of appearance models in visual object tracking," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, article 58, 2013.

[7] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[8] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," *Advances in Neural Information Processing Systems*, pp. 100–110, 2000.

[9] M. Karasuyama and I. Takeuchi, "Multiple incremental decremental learning of support vector machines," *IEEE Transactions on Neural Networks*, vol. 21, no. 7, pp. 1048–1059, 2010.

[10] F. Lorenzelli and K. Yao, *Systolic Arrays for SVD Downdating, SVD and Signal Processing III: Algorithms, Architectures, and Applications*, Elsevier Science, 1995.

[11] P. Hall, D. Marshall, and R. Martin, "Adding and subtracting eigenspaces with eigenvalue decomposition and singular value decomposition," *Image and Vision Computing*, vol. 20, no. 13-14, pp. 1009–1016, 2002.

[12] P. Hall, D. Marshall, and R. Martin, "Merging and splitting eigenspace models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 9, pp. 1042–1049, 2000.

[13] D. Skočaj and A. Leonardis, "Weighted and robust incremental method for subspace learning," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, vol. 2, pp. 1494–1501, Nice, France, October 2003.

[14] A. Levy and M. Lindenbaum, "Sequential karhunen-loeve basis extraction and its applicat ion to images," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1371–1374, 2000.

[15] D. Huang, Z. Yi, and X. Pu, "A new incremental PCA algorithm with application to visual learning and recognition," *Neural Processing Letters*, vol. 30, no. 3, pp. 171–185, 2009.

[16] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1–3, pp. 125–141, 2008.

[17] J. Weng, Y. Zhang, and W. Hwang, "Candid covariance-free incremental principal component analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 1034–1040, 2003.

[18] H. Zha and H. D. Simon, "On updating problems in latent semantic indexing," *SIAM Journal on Scientific Computing*, vol. 21, no. 2, pp. 782–791, 1999.

[19] A. Levy and M. Lindenbaum, "Sequential Karhunen-Loeve basis extraction and its application to images," in *Proceedings of the International Conference on Image Processing (ICIP '98)*, vol. 2, pp. 456–460, Chicago, Ill, USA, October 1998.

[20] H. Zhao, P. C. Yuen, and J. T. Kwok, "A novel incremental principal component analysis and its application for face recognition," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 36, no. 4, pp. 873–886, 2006.

[21] M. Brand, "Fast low-rank modifications of the thin singular value decomposition," *Linear Algebra and its Applications*, vol. 415, no. 1, pp. 20–30, 2006.

[22] J. Melenchón and E. Martínez, "Efficiently downdating, composing and splitting singular value decompositions preserving the mean information," in *Proceedings of the 3rd Iberian Conference on Pattern Recognition and Image Analysis*, pp. 436–443, 2007.

[23] L. Wang, S. Chen, and C. Wu, "An accurate incremental principal component analysis method with capacity of update and downdate," *Proceedings of International Conference on Computer Science & Information Technology*, vol. 51, pp. 118–123, 2012.

[24] M. Artac, M. Jogan, and A. Leonardis, "Incremental pca for online visual learning and recognition," in *Proceedings of the 16th International Conference on Pattern Recognition*, vol. 3, pp. 781–784, 2002.

[25] P. Jonathon Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[26] A. M. Martinez and R. Benavente, "The AR face database," CVC Technical Report 24, 1998.

[27] 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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[28] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-100)," Tech. Rep. CUCS-006-96, 1996.

[29] S. Ozawa, S. Pang, and N. Kasabov, "Incremental learning of chunk data for online pattern classification systems," *IEEE Transactions on Neural Networks*, vol. 19, no. 6, pp. 1061–1074, 2008.

[30] N. Goel, G. Bebis, and A. Nefian, " Face recognition experiments with random projection," in *2nd Biometric Technology for Human Identification*, vol. 5779 of *Proceedings of SPIE*, pp. 426–437, Orlando, Fla, USA, March 2005.

[31] S. Dasgupta and A. Gupta, "An elementary proof of a theorem of Johnson and Lindenstrauss," *Random Structures & Algorithms*, vol. 22, no. 1, pp. 60–65, 2003.