Received: 9 February 2020 | Revised: 18 January 2021 | Accepted: 14 February 2021

DOI: 10.1002/nla.2370

RESEARCH ARTICLE

WILEY

Summation pollution of principal component analysis and an improved algorithm for location sensitive data

Jingwei Li[1] | **Xiao-Chuan Cai**[1,2]

[1]Department of Computer Science, University of Colorado Boulder, Boulder, Colorado, USA

[2]Department of Mathematics, University of Macau, Macau, China

Correspondence
Xiao-Chuan Cai, Department of Computer Science, University of Colorado Boulder, Boulder, CO, USA.
Email: xccai@um.edu.mo

Funding information
NSF, Grant/Award Number: DMS-1720366

Abstract

Principal component analysis (PCA) is widely used for dimensionality reduction and unsupervised learning. The reconstruction error is sometimes large even when a large number of eigenmode is used. In this paper, we show that this unexpected error source is the pollution effect of a summation operation in the objective function of the PCA algorithm. The summation operator brings together unrelated parts of the data into the same optimization and the result is the reduction of the accuracy of the overall algorithm. We introduce a domain decomposed PCA that improves the accuracy, and surprisingly also increases the parallelism of the algorithm. To demonstrate the accuracy and parallel efficiency of the proposed algorithm, we consider three applications including a face recognition problem, a brain tumor detection problem using two- and three-dimensional MRI images.

KEYWORDS

dimensionality reduction, domain decomposition, image recognition, parallel computing, principle component analysis, subspace optimization

1 | INTRODUCTION

Principal component analysis (PCA) is a widely used tool for dimensionality reduction and unsupervised learning.[1] Its accuracy can be measured by the reconstruction error which is the norm of difference between the low-dimensional approximation projected back to the original high dimensional space and the original data. When the error is large, people often think that the dimension of approximation space is too low. In this paper, we point out that there is another cause of error that is the cross-summation pollution inherited in the formulation of PCA itself. Such a pollution issue is also observed in a recent paper for cancer detection using genetic data[2] in which the authors found that sometimes more data implies more error and the results calculated in a subspace using less data is actually better. The other issue with PCA is the computational cost of the singular value calculation which is high for large dataset and difficult to parallelize on supercomputers with a large number of processors. To deal with both issues, we propose and study a domain decomposed version of PCA that is more accurate under certain assumptions and also highly parallel.

We briefly survey some recent development of PCA for large-scale problems for which several distributed versions of PCA are developed in order to map the data and the computation on to distributed memory parallel computers.[3-5] In Reference 6, the author introduced a process that first decomposes the problem into blocks using a priori knowledge of the dataset, and then using a distributed strategy based on MapReduce to compute the principal components for each block. Balcan et al.[7] studied a distributed PCA which leads to lower communication and computational costs for the k-means clustering and related problems. In Reference 8, a simple and effective method referred to as the

transformation of a data matrix to a Gaussian matrix was suggested. This transformed data matrix is more suitable for PCA and independent component analysis. Sommer[9] created a data filtering method for the parallel k-NN search based on PCA, which is highly scalability for a wide range of high-dimensional dataset on multicore platforms. In Reference 10, the authors proposed a sparse contrastive PCA capable of handling sparse, interpretable, stable, and pertinent biological signal. More recently the authors[11] applied successfully of PCA in a multivariate time-correlated linear process.

PCA has also been extensively used in industry. For example, Reference 12 develops a new framework for data compression in seismic sensor networks by using a distributed PCA, which compresses all seismic traces in the network at the sensor level. Reference 13 develops a robust PCA method that can pursue and remove outliers, exactly recover a low-rank matrix and calculate the optimal mean using a $l_{2,1}$-norm-based loss function and a Schatten p-norm regularization term. Reference 14 introduces a reliability analysis, as well as a way to judge the quality of the results. Reference 15 reduces the influence of grosses like variations in lighting, facial expressions, and occlusions to improve the robustness of PCA and presents a simple but effective unsupervised preprocessing approach for two-dimensional whitening reconstruction. Reference 9 provides a probabilistic and infinitesimal view of how the PCA procedure can be generalized to analyze the nonlinear manifold valued data that does not resort to linearization of the data space. Reference 16 proposes a framework for frequency-dependent PCA, which facilitates Priestley process-based simulation of multicomponent ground motions. Reference 17 presents a wide-area monitoring method to detect and locate power system disturbances by PCA and the k-nearest neighbor analysis. Reference 18 employs PCA to extract the turbulent part of the spectroscopic cubes for velocity gradient calculation. Reference 19 introduces a PCA-aided optimization technique to solve the multi-echelon biomass supply chain problem with the consideration of economic, environmental, and social dimensions. Reference 20 gives an overview of recent developments of PCA when the data is incomplete, especially, with specific missing data processes (i.e., ignorable and nonignorable mechanisms).

The rest of the paper is organized as follows. We first discuss the summation pollution issue of the classical principle component analysis in Section 2. Then we introduce a domain decomposed PCA that improves the accuracy, and also increases the parallelism of the algorithm in Section 3. A simple analysis is also provided in this section. To demonstrate the accuracy and parallel efficiency of the proposed algorithm, in Section 4, we consider some applications including a face recognition problem, and a brain tumor detection problem using MRI images. Finally we make some concluding remarks in Section 5.

2 | SUMMATION POLLUTION OF THE CLASSICAL PCA

The goal of PCA is to find the best low rank approximation of a given matrix. Suppose we are given a dataset $\{x_k \in \mathbb{R}^m | k = 1, \dots, n\}$, which can be assembled as a matrix

$$X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{m\times n}. \tag{1}$$

PCA computes an orthonormal matrix $V \in \mathbb{R}^{m\times d}$, where d is an integer much smaller than m such that $\{y_k = V^T x_k \in \mathbb{R}^d, k = 1, \dots, n\}$ forms a reduced dimensional space that keeps important features of X and the variance of the projected vectors is maximized. We define

$$H_{m\times d} = \left\{ V \;\middle|\; V \in \mathbb{R}^{m\times d}, V^T V = I_{d\times d} \right\},$$

where $I_{d\times d}$ is an $d \times d$ identity matrix, and

$$J(V) = \sum_{k=1}^{n} \left\| y_k - \frac{1}{n}\sum_{l=1}^{n} y_l \right\|_2^2 = \sum_{k=1}^{n} \left\| V^T \left(x_k - \frac{1}{n}\sum_{l=1}^{n} x_l \right) \right\|_2^2. \tag{2}$$

PCA is to find V that solves the following optimization problem

$$\max_{V\in H_{m\times d}} J(V). \tag{3}$$

Let the mean of the dataset be $\mu = 1/n \sum_{l=1}^{n} x_l \in \mathbb{R}^m$. We denote the "centered" data as $\hat{x}_k = x_k - \mu$, and $\hat{X} = [\hat{x}_1, \ldots, \hat{x}_n]$. To obtain the matrix V, we first compute the singular value decomposition (SVD) of $\hat{X}$ as follows:

$$\hat{X} = \hat{U}\hat{\Sigma}\hat{W}^T, \tag{4}$$

where $\hat{U}$ is an $m \times m$ orthogonal matrix, $\hat{\Sigma}$ is an $m \times n$ diagonal matrix of singular values $\sigma_1, \ldots, \sigma_n$ arranged in a decreasing order, and $\hat{W}$ is an $n \times n$ orthogonal matrix. The solution to the optimization problem (3) is $V = \hat{U}_d$, consisting of the first d left singular vectors of $\hat{X}$. Let $\hat{\Sigma}_d$ be the top left $d \times d$ block of $\hat{\Sigma}$, and $\hat{W}_d$ the first d columns of $\hat{W}$, we obtain the projected matrix as

$$Y = \hat{U}_d^T \hat{X} = \hat{\Sigma}_d \hat{W}_d^T \in R^{d \times n}, \tag{5}$$

which is a low-dimensional representation of X. Each column of Y, say y_k, can be used to reconstruct an approximation corresponding x_k denoted as

$$\tilde{x}_k = V y_k.$$

The reconstruction error is measured by

$$\epsilon_k = ||x_k - \tilde{x}_k||_2.$$

In the reminder of the paper, we refer to this version of PCA as the global PCA, denoted as GPCA. Below we take a closer look of the objective function (2). Let

$$g_k = V^T \left(x_k - \frac{1}{n} \sum_{l=1}^{n} x_l \right). \tag{6}$$

GPCA can be described as:[1] Computes an orthogonal matrix $V \in \mathbb{R}^{m \times d}$ such that maximizes the following objective function $J(V)$

$$\max_V J(V) = \sum_{k=1}^{n} \left|\left| g_k \right|\right|_2^2. \tag{7}$$

Consider the situation when m is large, each vector x_k can be partitioned into p subvectors

$$x_k = \begin{bmatrix} x_k^1 \\ \vdots \\ x_k^p \end{bmatrix},$$

where $x_k^i \in \mathbb{R}^{m_i}$, and

$$\sum_{i=1}^{p} m_i = m.$$

Compatibly we partition the rows of the matrix V as

$$V = \begin{bmatrix} V_1 \\ \vdots \\ V_p \end{bmatrix}_{m \times d},$$

FIGURE 1 The left figure is the original image, the middle figure is the reconstruction from principle component analysis (PCA) using 10 eigenmodes, and the right figure is obtained by splitting the image into nine subimages and then apply nine subspace PCAs with 10 eigenmodes

where $V_i \in \mathbb{R}^{m_i \times d}$. Then, we have

$$\left\| V^T \left(x_k - \frac{1}{n} \sum_{l=1}^{n} x_l \right) \right\|_2^2 = \left\| \sum_{i=1}^{p} V_i^T \left(x_k^i - \frac{1}{n} \sum_{l=1}^{n} x_l^i \right) \right\|_2^2. \tag{8}$$

We partition g_k defined in (6) into p subvectors, then each subvector can be written as

$$g_k^i = V_i^T \left(x_k^i - \frac{1}{n} \sum_{l=1}^{n} x_l^i \right), \quad i = 1, \ldots, p,$$

then (8) becomes

$$\left\| \sum_{i=1}^{p} g_k^i \right\|_2^2.$$

Thus, a partitioned version of GPCA takes the form:

$$\max_{V \in \mathbb{R}^{m \times d}} J(V) = \sum_{k=1}^{n} \left\| \sum_{i=1}^{p} g_k^i \right\|_2^2. \tag{9}$$

In (9), the outer summation $\sum_{k=1}^{n}$ is easy to understand since it goes over all n sample vectors. The inner summation $\sum_{i=1}^{p}$ is not always meaningful. For example, in a face recognition problem each x_k (with g_k as its lower-dimensional approximation) is a picture of a face as shown in the left figure of Figure 1. If the picture is partitioned into nine subpictures $(g_k^1, \ldots g_k^9)$ as shown in the right figure of Figure 1, the inner summation simply takes the sum of these nine subpictures that are irrelevant to each other. It might be meaningful to keep some nearby subvectors in the same summation since they may impact certain features of the figure, but the pieces that are far away from each other should not be in the same summation since they pollute each other and lower the overall accuracy of the algorithm.

A better approach would be: For each $1 \le i \le p$, we compute an orthogonal matrix $W_i \in \mathbb{R}^{m_i \times d}$ such that

$$\max_{W_i} J(W_i) = \sum_{k=1}^{n} \left\| g_k^i \right\|_2^2, \tag{10}$$

without the inner summation in (9), and then form a global orthogonal matrix by stacking and rescaling of $W_1, \ldots, W_p$. The detailed definitions of g_k^i and W_i will be given in the next section.

In Figure 1, we show an application of the above-mentioned idea for a face recognition problem. The left figure is the original image, the middle figure is the reconstruction from PCA using 10 eigenmodes, and the right figure is obtained by splitting the image into nine subimages and then apply nine subspace PCAs. It is clear that the right figure is better than the middle one. The effect is more pronounced for larger scale problems.

Borrowing a term from numerical methods for partial differential equations,[21-25] we refer to method (10) as the domain decomposed PCA (DDPCA) in the rest of the paper. The method avoids cross pollution from unrelated parts of the data

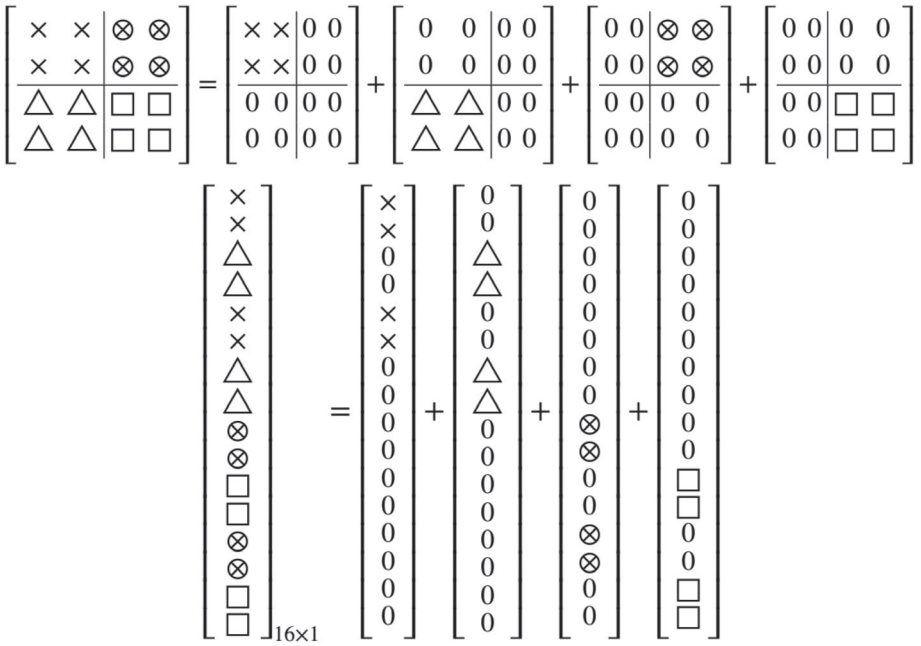

FIGURE 2 A partition of a 4×4 figure into four subfigures in the matrix form (in the top figure) and in the vector form (bottom figure)

by removing the inner summation operator in (9). An added benefit is that these subspace optimization problems (10) are independent of each other, and can be solved in parallel.

Depending on the formulation, PCA requires the computation of singular values (or eigenvalues) of a matrix which is often large and dense. It is quite a challenge to parallelize the singular value solver for use on cluster of computers, or large-scale supercomputers.[26] This limits the applicability of the method to relatively small problems. This is often acceptable for two-dimensional applications, but not for three-dimensional applications. Our proposed domain decomposed PCA approach maps the dataset naturally to parallel machines, and we show computationally that the parallel speedup is almost linear. More importantly, the new method is substantially more accurate than the classical PCA for the two application problems that we have considered including a two-dimensional face recognition problem, a three-dimensional brain tumor detection problem.

In the numerical experiments section we show that the new version is more accurate and also faster when implemented on a parallel computer since the subspace computations can all be carried out independently.

3 | SOME ANALYSIS OF THE DDPCA

By introducing a decomposition of the location space into subspaces, we have a decomposition of all the vectors in the sample space. We then solve the problems defined in the subspaces separately in parallel. A global solution is obtained by combining the subspace solutions with an appropriate ordering and scaling. In Figure 2, we show a partition of a 4×4 image into four subfigures in the matrix form (in the top figure) and in the vector form (bottom figure) when the pixels of the image are ordered from the top to the bottom and from the left to the right. If we consider the original image as a vector in $\mathbb{R}^{16}$, then the subfigures are in four orthogonal subspaces of dimension 4 embedded in $\mathbb{R}^{16}$.

Let p be the number of subspaces. There are many ways to partition the whole space into p subspaces, and the best partition is usually application dependent. To define the general algorithm, for each $1 \le i \le p$, we define a "global to local" mapping operator $I_g^i : x_k \to x_k^i$, in order to generate the subspace vectors. More precisely speaking, we define a subidentity matrix of size $m_i \times m$ and its entries are either 0 or 1 such that

$$x_k^i = I_g^i x_k \in \mathbb{R}^{m_i}, \quad \text{where} \quad k = 1, \dots, n \quad \text{and} \quad i = 1, \dots, p,$$

and $\Sigma_{i=1}^{p} m_i = m$. In other words, the product of I_g^i with a global image returns the ith subfigure. Recall that d is the desired dimension of the reduced space, let us define the ith subspace of orthogonal matrices

$$H_{m_i\times d} = \left\{ W_i \;\middle|\; W_i \in \mathbb{R}^{m_i\times d}, W_i^T W_i = I_{d\times d} \right\}.$$

For the samples $x_1^i, ..., x_n^i$, we introduce a subspace objective function

$$J_i(W_i) = \sum_{k=1}^{n} \left|\left| W_i^T \left(x_k^i - \frac{1}{n}\sum_{l=1}^{n} x_l^i \right) \right|\right|_2^2.$$

Then the domain decomposed PCA reads: for each i, computes a W_i that solves the following subspace optimization problem

$$\max_{W_i \in H_{m_i\times d}} J_i(W_i). \tag{11}$$

Once the subspace problems are solved, we define a global projection matrix

$$W = \sum_{i=1}^{p} (I_g^i)^T W_i,$$

it is easy to see that

$$W^T W = \sum_{i=1}^{p} W_i^T I_g^i (I_g^i)^T W_i = \sum_{i=1}^{p} W_i^T W_i = pI_{d\times d},$$

since $I_g^i (I_g^i)^T = I_{m_i\times m_i}$ for $i = 1 \ \dots \ p$ and $I_g^i (I_g^j)^T = 0_{m_i\times m_i}$ for $i \neq j$ and $i, j = 1, \ \dots \ p$. Finally, we scale the matrix W to define the global projection matrix

$$\tilde{V} = \frac{1}{\sqrt{p}} W,$$

which is the solution of the global problem.

To compare with GPCA, let

$$\tilde{g}_k^i = W_i^T \left(x_k^i - \frac{1}{n}\sum_{l=1}^{n} x_l^i \right),$$

then for the ith subproblem in DDPCA, the objective function in (11) takes the form

$$\max \sum_{k=1}^{n} \left|\left| \tilde{g}_k^i \right|\right|_2^2,$$

which is simply the objective function (9) in GPCA without the inner summation. Below we make some remarks about the new version of PCA.

Remark 1. DDPCA is not intended for small problems; i.e., small m. The number of subspaces p is usually much smaller than the dimension m of the sample vectors so that each subvector contains sufficient features of interests. In our applications, we simply pick p as the number of processors of the parallel computer.

Remark 2. For a chosen p there are many ways to partition the vector x_k into $x_k^1, \ \dots \ , x_k^p$, and whenever possible we use the location information so that the components of each x_k^i are related. This is easy to do for images but may not be clear for other types of data for which the ordering of the components of the vector is arbitrary.

Remark 3. The partition is applied to each sample vector, and we do not introduce a partition among the sample vectors. The algorithm is intended for situations where x_k are long vectors, and not for dataset with a large number of short vectors.

Next we provide some analysis of the proposed DDPCA based on the decays of the singular values. The analysis is motivated with an observation from several classes of practical problems that we have been working on.

Observation: Suppose that the global matrix has a SVD as in (4), and as discussed above we partition all the images into p subimages, and each $\hat{X}^i \in \mathbb{R}^{m_i \times n}$ has a SVD as following

$$\hat{X}^i = U^i \Sigma^i (V^i)^T \quad \text{for} \quad i = 1, \ldots, p,$$

where $U^i \in \mathbb{R}^{m_i \times m_i}$, $\Sigma^i \in \mathbb{R}^{m_i \times n}$, and $V^i \in \mathbb{R}^{n \times n}$. Let $\sigma_l^i, l = 1, \ldots$ be the singular values of X^i arranged in a decreasing order.

Then, there exists a $q > 0$, for $l \geq q$,

$$\sigma_l^i \leq \frac{1}{\sqrt{p}} \sigma_l \quad \text{for} \quad i = 1, \ldots, p, \tag{12}$$

where q is the smallest number of the eigenmodes we need to compute, σ_l^i is the lth largest singular value of $\hat{X^i}$, and σ_l is the lth largest singular value of $\hat{X}$. Equation (12) says that the singular values of the submatrices decay not only faster than that of the global matrix, but actually faster by a factor that is strictly less than 1. q does not appear explicitly in (12) but it is the lower bound for l. In other words, q is the smallest number of eigenmodes such that (12) holds.

We further observe that for images with distinctive features, the value of q is often small. For some of problems to be presented in the numerical experiment section of the paper, we computed the q values. For example, for the face recognition problem $q = 3$ (when $p = 4$). For the 2D brain tumor problem $q = 14$ (when $p = 4$). For the 3D brain tumor problem, $q = 23$ (when $p = 4$). On the other hand, for featureless images, the value of q can be large. For example, if we fix $p = 4$, when the images (1024×1024) are generated by random numbers with a normal distribution, then $q = 148$, and it becomes $q = 297$ when the random numbers are generated with a uniform distribution function.

Theorem. *If the dimension d of the reduced space is q or larger, then the reconstruction residual of DDPCA is smaller than the reconstruction residual of GPCA.*

Proof. Let $\hat{X}_i = I_g^i \hat{X}$, then we have

$$\left|\left|\hat{X}\right|\right|_F^2 = \sum_{i=1}^{p} \left|\left|\hat{X}_i\right|\right|_F^2.$$

Consider the SVD of $\hat{X}$ and $\hat{X}^i$, and by the Eckart–Young–Mirsky theorem,[27] we have the reconstruction residual of GPCA

$$\text{resid}_{\text{GPCA}}^2 = \left|\left|\hat{X} - \hat{X}_d\right|\right|_F^2 = \sum_{k=d+1}^{\min(m,n)} \sigma_k^2,$$

and the reconstruction residual of DDPCA

$$\text{resid}_{\text{DDPCA}}^2 = \sum_{i=1}^{p} \left|\left|\hat{X}^i - \sum_{i=1}^{p} \hat{X}_d^i\right|\right|_F^2 = \sum_{i=1}^{p} \sum_{k=d+1}^{\min(m_i,n)} \left(\sigma_k^i\right)^2.$$

Since for sufficiently large d (i.e., $d \geq q$),

$$\sigma_k^i \leq \frac{1}{\sqrt{p}} \sigma_k, \quad k = d+1, \ldots, \min(m_i, n),$$

we have

$$\text{resid}_{\text{DDPCA}}^2 \leq \sum_{i=1}^{p} \sum_{k=d+1}^{\min(m,n)} \left(\frac{1}{\sqrt{p}} \sigma_k\right)^2 = \sum_{i=1}^{p} \frac{1}{p} \left(\sum_{k=d+1}^{\min(m,n)} \sigma_k^2\right) = \text{resid}_{\text{GPCA}}^2.$$

■

4 | NUMERICAL EXPERIMENTS

In this section we present some numerical experiments to show the superiority of the proposed DDPCA over the classical PCA for two classes of problems including a face recognition problem, and a brain tumor detection problem. For the brain tumor problem we consider both two-dimensional and three-dimensional MRI images. Since the three-dimensional images are quite large, we have implemented the algorithm in parallel and will present results obtained on a supercomputer with over 1000 processors. The input dataset is represented as vectors $x_k,\ k=1,\ \dots\ ,n$, and $\tilde{x}_k$ is the corresponding reconstructed vector. To measure the accuracy, we compute the reconstruction error $\epsilon_k = ||x_k - \tilde{x}_k||_2$, for $k=1,\ \dots\ ,n$.

4.1 | Face recognition

This experiment is based on the "ORL Database of Faces," which is updated by AT&T Laboratories, Cambridge UK.[28] There are 40 distinct persons and each person has 10 different images. The gray scale images are of dimension 112×92. The total number of pixels per image is 10,304, which is the dimension of each x_i.

Figure 3 shows the GPCA and DDPCA reconstruction of the images. The top 4 figures show the reconstructions from the GPCA algorithm with 1, 10, 50, and 150 eigenmodes. It is clear that using more eigenmodes gives more accurate results. The other three rows of Figure 3 are the reconstructions from DDPCA with 2×2, 3×3, and 4×4 partitions, respectively, , and also using 1, 10, 50, and 150 eigenmodes. Except the case when the number of eigenmode is 1, the results from DDPCA are much better than that of GPCA corresponding to the same number of eigenmode.

To quantitatively understand the accuracy of the algorithms, in Figure 4, we plot the reconstruction errors of GPCA and DDPCA with different partitions for one of the images in the dataset, as shown in Figure 1, using up to 200 eigenmodes. The errors for other images in the dataset are more or less the same. As we can see, when the number of eigenmode is fixed, the error from the GPCA algorithm is greater than that of DDPCA. For GPCA, the error stops decreasing after

FIGURE 3 The top row shows the reconstructions computed with global principle component analysis (PCA) using 1, 10, 50, 150 eigenmodes; the other three rows of figures show the reconstructions computed with domain decomposed PCA with 2×2, 3×3, 4×4 partitions using the same number of eigenmodes, respectively

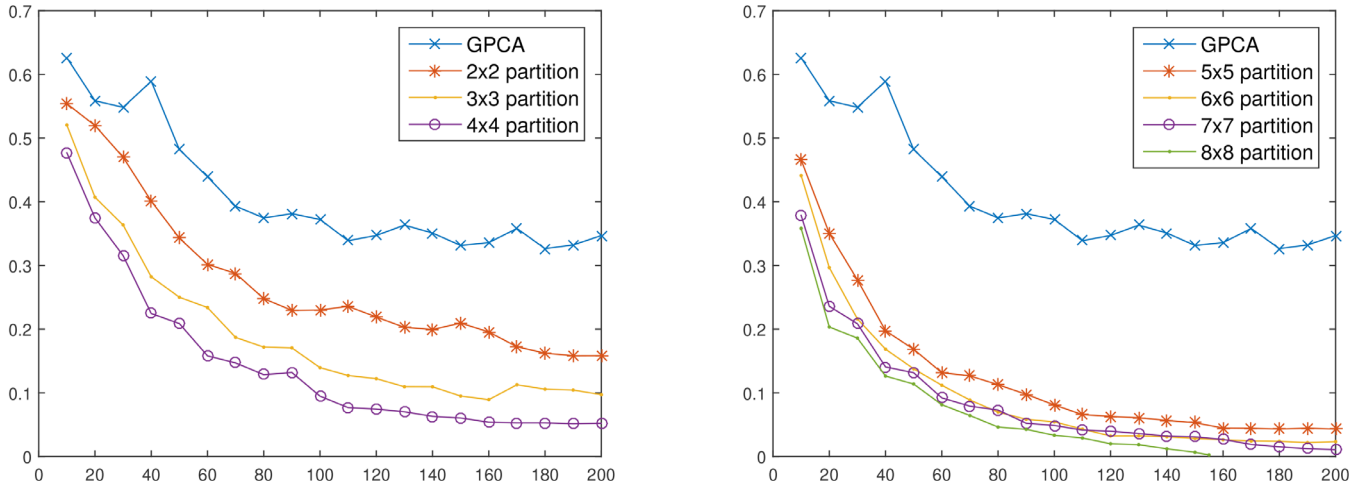

FIGURE 4 Reconstruction error of the face recognition problem (vertical axis) with global principle component analysis (PCA) and domain decomposed PCA with different partitions. The horizontal axis is the number of eigenmodes

a certain number of eigenmodes; in other words, there is no convergence. On the other hand, for DDPCA, the error decreases as we increase the number of eigenmodes. Surprisingly, when we fix the number of eigenmodes and increase the number of subdomains, we obtain much more accurate results. Note that when running the code on a parallel machine the total compute time per processor decreases linearly as we increase the number of subdomains. This put DDPCA in a total advantage comparing with GPCA since it is not only more accurate, but also faster when implemented on parallel computers.

4.2 | Brain tumor detection from MRI images

This experiment is based on a brain tumor MRI database, the brain T1-weighed CE-MRI dataset, acquired by Nanfang Hospital and Tianjing Medical University.[29] In this experiment, we select 51 patients from the CE-MRI dataset and the size of the 3D image is $512 \times 512 \times 6$. We perform two experiments using the dataset; a 2D experiment based on 2D slices of the images, and a 3D experiment based directly on the 3D images. Each 2D slice has 262,144 pixels and each 3D image has 1,572,864 pixels. The dataset consists of three types of brain tumors: meningioma, glioma, and pituitary tumor. In the end of the section we provide a classification of the tumors using the resulting low-dimensional data from the PCA algorithm.

We first consider slices of images. The third dimension is not used in the sense that the connections between the slices are ignored. The 2D detection is much cheaper than the 3D detection, and it assumes that if one slice has a tumor then the patient has a tumor, however, sometimes if the tumor is vertically large but looks small in all the slices then the tumor might be missed by the 2D algorithm.

In Figure 5, we show the reconstruction results obtained using GPCA and DDPCA with 2×2, 8×8, and 32×32 partitions. The top row is from GPCA with 1, 10, 50, 150 eigenmodes, and the other three rows of figures are from DDPCA calculations with the same number of eigenmodes. One can see clearly that GPCA requires at least 50 eigenmodes to find the initial shape of the tumor, but DDPCA is able to find it with only one eigenmode when a 32×32 partition is used.

As far as we know all the existing applications of PCA for brain tumor detection are restricted to 2D slices since 3D calculation is too expensive. The left figure of Figure 6 is the 3D image of one patient and the right figure shows a zoom-in of the tumor. When applying DDPCA to 3D images, we do not decompose the third dimension since the number of slices is small for this particular dataset.

Figure 7 shows the reconstruction with 45 eigenmodes. The top left figure on the first row is from GPCA. It is clear that the result is not acceptable. The other three figures on the first row are the results of DDPCA with $4 \times 4 \times 1$, $8 \times 8 \times 1$, and $16 \times 16 \times 1$ decomposition, and the accuracy is quite reasonable. The second row is a zoom-in version of the figures on the first row near the tumor, respectively.

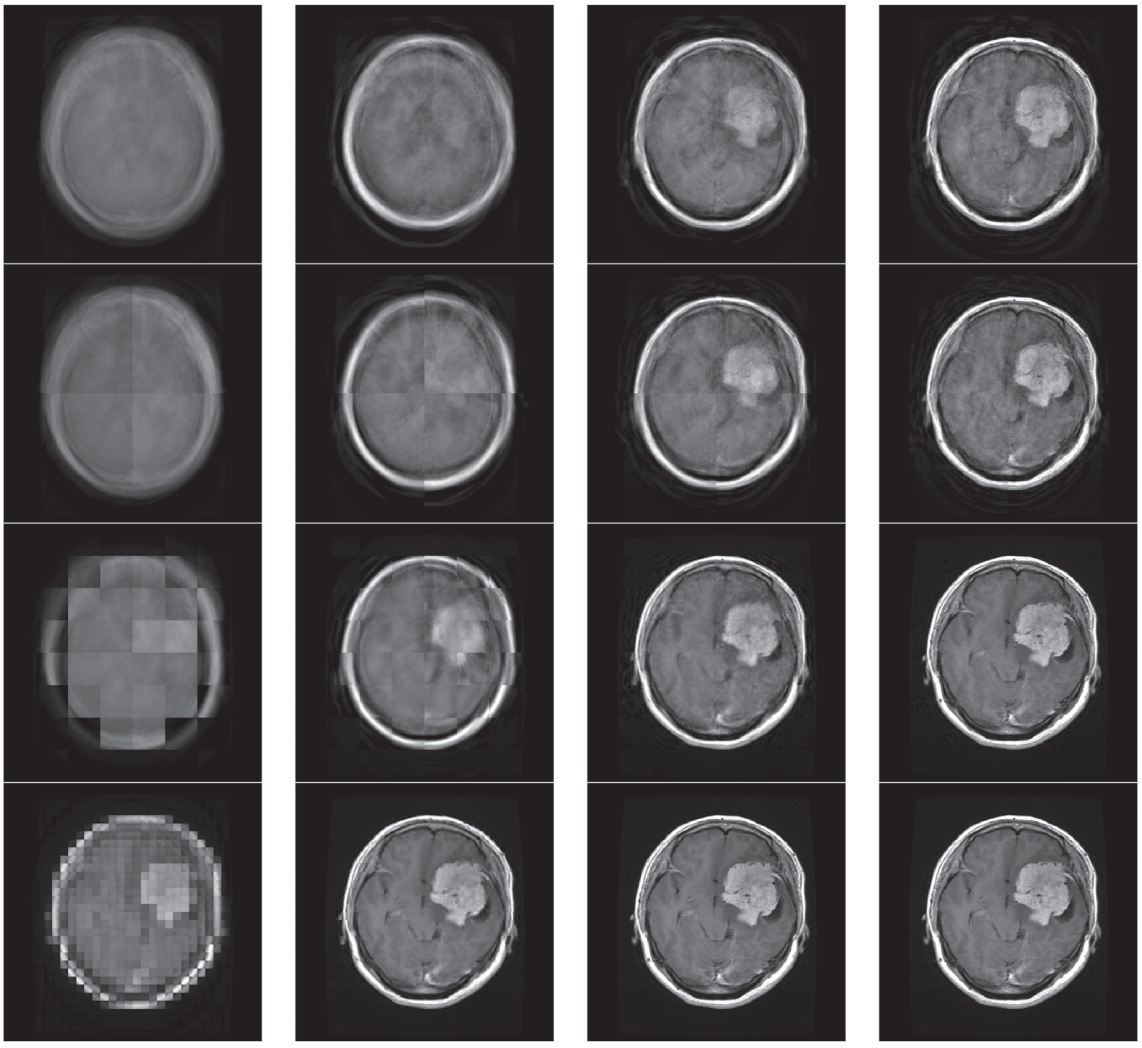

FIGURE 5 The top four figures are the reconstructions from global principle component analysis (PCA) with 1, 10, 50, and 150 eigenmodes; the other three rows are from domain decomposed PCA with a 2×2, 8×8, 32×32 decomposition using the same number of eigenmodes, respectively

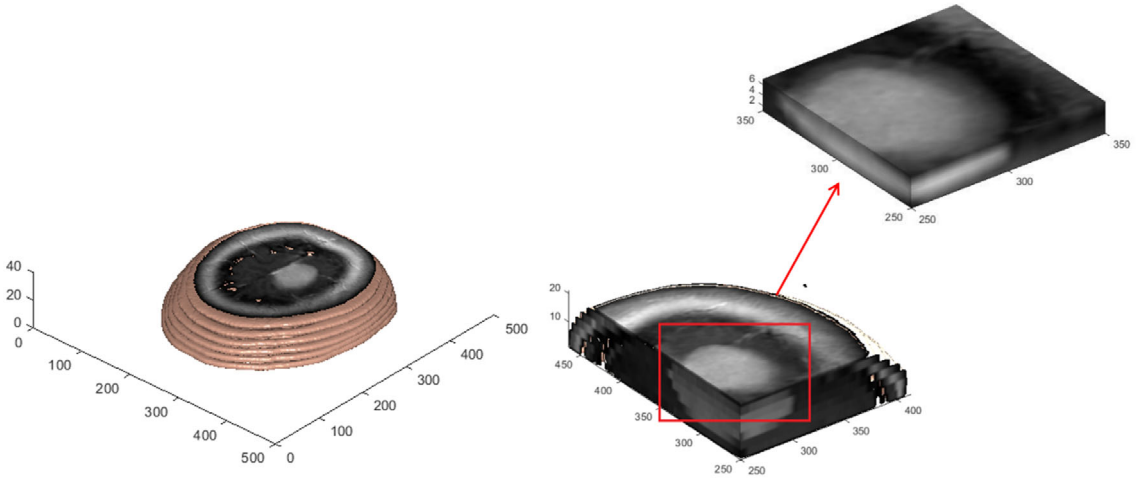

FIGURE 6 Left is the original three-dimensional image; right is a zoom-in including part of the tumor

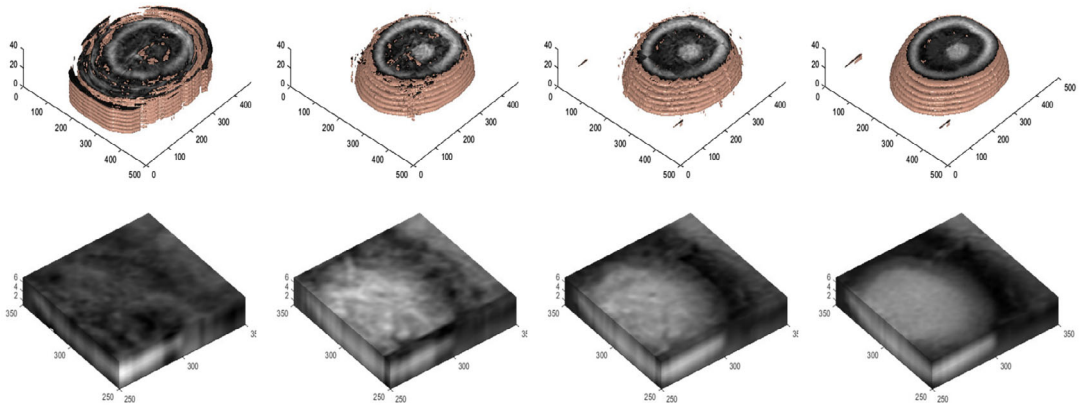

FIGURE 7 The left figure on the first row is the reconstruction using global principle component analysis (PCA); the other three figures on the first row are from three-dimensional domain decomposed PCA with different decompositions 4×4, 8×8, and 16×16. The second row is a zoom-in of the figures on the first row near the tumor, respectively

Figure 8 shows reconstruction errors for various number of eigenmodes using GPCA and DDPCA with different domain decompositions for the patient whose image is shown in Figure 6. In this experiment, $m = 512 \times 512 \times 6 = 1,572,864$, $n = 51$. Since the computations are expensive, we only use d up to 50 eigenmodes. The error of the GPCA calculation is quite large, while DDPCA with larger number of subdomains has basically converged.

In Figure 9, we show some results obtained on a parallel computer using up to 1024 processors. The figure shows the total compute time and the reconstruction errors for GPCA and DDPCA with several different partitions. The problem defined on each subdomain is allocated to a processor. The blue bar indicates the reconstruction error and the number above the bar is the corresponding compute time. The horizontal axis includes the number of processors and the decomposition used in DDPCA and GPCA corresponds to the no partition case. As shown clearly that DDPCA outperforms GPCA, by a huge margin, in terms of both the compute time and the accuracy.

Finally, we show the classification of brain tumors. Since this is not the main focus of this paper, we only briefly mention it as a potential application of the DDPCA algorithm. After the dimension reduction with PCA, each image x_i is represented by a lower-dimensional vector y_i. We then use a linear discriminant analysis algorithm[1] to classify the three types of brain tumors. Based on 2D brain tumors detection, we use totally 51 slices (17 slices for each type of tumors) and overall 50 patients in this experiment. In the left picture of Figure 10, we show the results obtained using GPCA computed with 15 eigenmodes which contains some classification error between meningioma and glioma. The right figure of Figure 10 shows the classification results of DDPCA computed with 15 eigenmodes on a 4×4 partition and they are quite satisfying.

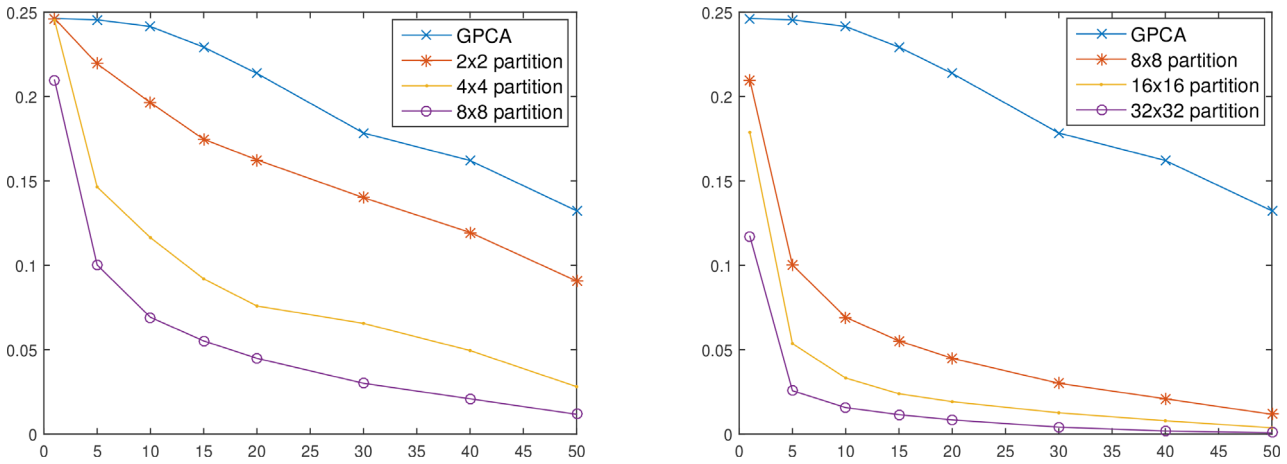

FIGURE 8 The reconstruction error using global principle component analysis (PCA) and domain decomposed PCA with 2×2, 4×4, 8×8, 16×16, 32×32 decompositions. The horizontal axis is the number of eigenmodes

FIGURE 9 The blue bar is the reconstruction error (vertical axis), and the number above the blue bar is the total compute time (in seconds). The left figure is for the face recognition problem and the right figure is for the two-dimensional brain tumor problem. The horizontal axis is the partition and the number of processors (*np*)

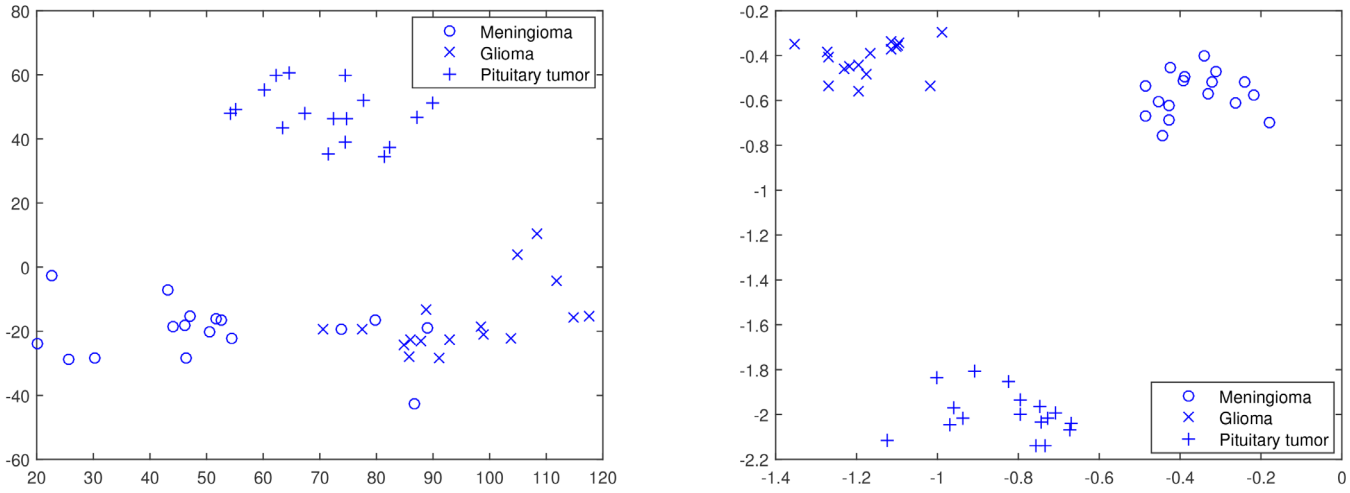

FIGURE 10 The left figure is classification results using global principle component analysis (PCA) with 15 eigenmodes; the right figure is the classification results using the domain decomposed PCA with a 4×4 decomposition and 15 eigenmodes

4.2.1 | Weak scalability

In this section, we test the weak scalability of DDPCA. We consider a 4×4 decomposition of the 2D brian tumor detection problem as an example. The matrix size is $16,384 \times 51$. If we use 16 processors to solve the problem, the total compute time is 5.449 seconds. Next we only solve 4 of the subproblems, that is, the matrix size is 4096×51 and use four processors, the total compute time is 5.732 s. Lastly, we solve only one of the subproblems and use one processor, the total compute time is 5.683 s. The three compute times, 5.449, 5.732, 5.683 are very close to each other, therefore we conclude that the method is weakly scalable.

5 | CONCLUSIONS

In this paper, we first discussed a mysterious property of PCA which we refer to as the summation pollution. This is a property that people have observed in some applications but have not connected it to the basic formulation of PCA itself. In other words, when using PCA to extract features from sample vectors that contain a large number of components with unrelated features, the accuracy is reduced because of the interference of these components imposed by the inner summation in the definition of the PCA algorithm. To avoid the pollution, we introduced a highly parallel, domain decomposed version of PCA, which is more accurate than the classical PCA in the sense that fewer eigenmodes are required to

achieve the same level of reconstruction error. A simple analysis is provided to relate the reconstruction residuals to the rates at which the singular values decrease in the global matrix and in the subdomain matrices. The algorithm is highly parallel since the problems defined on the subdomains are independent of each other and can be solved in parallel. As applications, we considered a face recognition problem, a 2D brain tumor detection problem, and a 3D brain tumor detection problem. We mention that the proposed algorithm has the potential for very large problems, and for supercomputers with a large number of processor cores.

ACKNOWLEDGEMENTS

The authors would like to thank the referees for insightful comments that helped improve this paper. This research is supported in part by NSF under DMS-1720366.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in ORL Database of Faces at http://www.uk.research.att.com/facedatabase.html, Reference 1.

ORCID

Xiao-Chuan Cai https://orcid.org/0000-0003-0296-8640

REFERENCES

1. Kokiopoulou E, Chen J, Saad Y. Trace optimization and eigenvalue problems in dimension reduction methods. Numer Linear Algebra Appl. 2010;18(3):565–602.
2. Lenz M, Muller F-J, Zenke M, Schuppert A. Principal components analysis and the reported low intrinsic dimensionality of gene expression microarray data. Sci Rep [Internet]. 2016;6(1):1–11. https://doi.org/10.1038/srep25696.
3. Fan J, Wang D, Wang K, Zhu Z. Distributed estimation of principal eigenspaces. Ann Stat. 2019;47(6):3009–31.
4. Feldman D, Schmidt M, Sohler C. Turning big data into tiny data: constant-size coresets for *k*-means, PCA and projective clustering. SIAM J Comput. 2020;49(3):601–57.
5. Qu Y, Ostrouchov G, Samatova N, Geist A. Principal component analysis for dimension reduction in massive distributed data sets. Paper presented at: Proceedings of the IEEE International Conference on Data Mining (ICDM), Maebashi City, Japan; 2002. p. 134–153.
6. Zhu J, Ge Z, Song Z. Distributed parallel PCA for modeling and monitoring of large-scale plant-wide processes with big data. IEEE Trans Ind Inform. 2017;13(4):1877–85.
7. Balcan M, Kanchanapally V, Liang Y, Woodruff D. Improved distributed principal component analysis. Proc Neur IPS. 2014;31(3):1063–71.
8. Singha A, Bhowmik MK. Enhancing performance of PCA, ICA through distribution transformation. Proceedings of the 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC); IEEE, 2017.
9. Sommer S. An infinitesimal probabilistic model for principal component analysis of manifold valued data. Sankhya Ser A. 2019;81(1):37–62.
10. Boileau P, Hejazi NS, Dudoit S. Exploring high-dimensional biological data with sparse contrastive principal component analysis. Bioinformatics. 2020;36(11):3422–30.
11. Zamprogno B, Reisen VA, Bondon P, Aranda Cotta HH, Reis NC Jr. Principal component analysis with autocorrelated data. J Stat Comput Simul. 2020;90(12):2117–35.
12. Liu B, Mohandes M, Nuha H, Deriche M, Fekri F. A distributed principal component analysis compression for smart seismic acquisition networks. IEEE Trans Geosci Remote Sens. 2018;56(6):3020–9.
13. Shi X, Nie F, Lai Z, Guo Z. Robust principal component analysis via optimal mean by joint $l_{2,1}$ and Schatten *p*-norms minimization. Neurocomputing. 2018;283:205–13.
14. Mooi E, Sarstedt M, Mooi-Reci I. Market research: the process, data, and methods using stata. Singapore, Asia: Springer; 2018.
15. Shi X, Guo Z, Nie F, Yang L, You J, Tao D. Two-dimensional whitening reconstruction for enhancing robustness of principal component analysis. IEEE Trans Pattern Anal Mach Intell. 2016;38(10):2130–6.
16. Das S, Hazra B. Frequency-dependent principal component analysis of multicomponent earthquake ground motions: frequency dependent PCA. Earthq Eng Struct Dyn. 2018;47(5):1360–6.
17. Cai L, Thornhill N, Kuenzel S, Pal B. Wide-area monitoring of power systems using principal component analysis and *k*-nearest neighbor analysis. IEEE Trans Power Syst. 2018;46(3):4913–23.
18. Hu Y, Yuen KH, Lazarian A. Improving the accuracy of magnetic field tracing by velocity gradients: principal component analysis. Mon Not R Astron Soc. 2018;480(1):1333–9.
19. Shen H, Loong L. Sustainability evaluation for biomass supply chain synthesis: novel principal component analysis aided optimization approach. J Clean Prod. 2018;189(1):941–61.

20. Geraci M, Farcomeni A. Principal component analysis in the presence of missing data. Advances in principal component analysis. Singapore, Asia: Springer; 2018. p. 47–70.
21. Cai X-C, Saad Y. Overlapping domain decomposition algorithms for general sparse matrices. Numer Linear Algebra Appl. 1996;3(3):221–37.
22. Cai X-C, Sarkis M. A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM J Sci Comput. 1999;21(2):792–7.
23. Luo L, Liu L, Cai X-C, Keyes DE. Fully implicit hybrid two-level domain decomposition algorithms for two-phase flows in porous media on 3D unstructured grids. J Comput Phys. 2020;409(109312):109312.
24. Smith B, Bjorstad PE, Gropp W. Domain decomposition: parallel multilevel methods for elliptic partial differential equations. Cambridge, UK: Cambridge University Press; 2004.
25. Toselli A, Widlund OB. Domain decomposition methods - algorithms and theory. Berlin, Germany: Springer; 2010.
26. Hernandez V, Roman JE, Vidal V. SLEP: a scalable and flexible toolkit for the solution of eigenvalue problems. ACM Trans Math Softw. 2005;31(3):351–62.
27. Golub GH, Hoffman A, Stewart GW. A generalization of the Eckart-Young-Mirsky matrix approximation theorem. Linear Algebra Appl. 1987;88–89:317–27.
28. Damkliang K. The ORL database of faces [Data file]. AT&T Laboratories Cambridge, UK: 2002. https://www.kaggle.com/kasikrit/att-database-of-faces. Accessed 06 Jan 2019.
29. Cheng J, Huang W, Cao S, et al. Correction: enhanced performance of brain tumor classification via tumor region augmentation and partition. PLoS One. 2015;10(12):e0144479.

How to cite this article: Li J, Cai X-C. Summation pollution of principal component analysis and an improved algorithm for location sensitive data. *Numer Linear Algebra Appl*. 2021;e2370. https://doi.org/10.1002/nla.2370