1D-PCA, 2D-PCA to nD-PCA

Hongchuan Yu and Mohammed Bennamoun
School of Computer Science and Software Engineering,
University of Western Australia, Perth, WA6009, Australia
E-mail: {yu, bennamou}@csse.uwa.edu.au

Abstract

In this paper, we first briefly reintroduce the 1D and 2D forms of the classical Principal Component Analysis (PCA). Then, the PCA technique is further developed and extended to an arbitrary n-dimensional space. Analogous to 1D- and 2D-PCA, the new nD-PCA is applied directly to n-order tensors \((n \geq 3)\) rather than 1-order tensors (1D vectors) and 2-order tensors (2D matrices). In order to avoid the difficulties faced by tensors computations (such as the multiplication, general transpose and Hermitian symmetry of tensors), our proposed nD-PCA algorithm has to exploit a newly proposed Higher-Order Singular Value Decomposition (HO-SVD). To evaluate the validity and performance of nD-PCA, a series of experiments are performed on the FRGC 3D scan facial database.

1. Introduction

Principal component analysis (PCA), which is also known as Karhunen-Loeve (KL) transform, is a classical statistic technique that has been applied to many fields, such as knowledge representation, face recognition and image compression. The objectives of PCA are to reduce the dimensionality of the dataset and identify new meaningful underlying variables. The key idea is to project the objects to an orthogonal subspace for their compact representations. It usually involves a mathematical procedure that transforms a number of correlated variables into a smaller number of uncorrelated variables, which are called principal components. The first principal component accounts for as much of the variability in the dataset as possible, and each succeeding component accounts for as much of the remaining variability as possible. Up to now, there has been an extensive literature that addresses both the theoretical aspect of the PCA method and its application aspects [2, 6, 5]. Nevertheless, it can be noted that in the classical 1D-PCA the 2D data sample (e.g. image) must be initially converted to a 1D vector form. The resulting sample vector will lead to a high dimensional vector space. It is consequently difficult to evaluate the covariance matrix accurately when the sample vector is very long and the number of training samples is small. Furthermore, it can also be noted that the projection of a sample on each principal orthogonal vector is a scale. Obviously, this will cause the sample data to be over-compressed. In order to solve this kind of dimensional problem, Yang et al. [7] proposed the 2D-PCA approach. The basic idea is to directly use 2D matrices to construct the corresponding covariance matrix instead of a 1D vector set. This improves the computational efficiency. Furthermore, it can be noted that the projection of a sample on each principal orthogonal vector is a vector. Thus, the problem of over-compression is alleviated in the 2D-PCA case.

In this paper, we will apply the PCA technique to a multi-dimensional dataset. This is mainly motivated by the rapid improvements of image processing in the last decade, such as medical MRI and CT datasets, and hyperspectral imagery datasets. The basic tool is a Higher Order Singular Value Decomposition (HO-SVD) [3], which will help us to deal with multiarray.

The remainder of this paper is organized as follows: In Section 2, some related works are briefly introduced. The nD-PCA is then developed in Section 3. The nD-PCA is performed on the FRGC 3D scan facial database [4] for multimodel face recognition in Section 4. Finally, some conclusions are given in Section 5.

2. Related Works

We assume that the readers are familiar with the classical PCA approach (with regard to 2D-PCA, please refer to [7]). Hereon, only the procedure of PCA is simply outlined as follows: (1) Construct a covariance matrix of a training sample set; (2) Perform Eigen Value Decomposition on it. The first kth eigenvectors span an eigen subspace; (3) Project a probe sample into this subspace for its compact representation or recognition.

With higher dimensional datasets, it is straightforward to use tensors to represent them. In our nD-PCA scheme,
a higher-order tensor based singular value decomposition (HO-SVD) algorithm [3] is exploited. For the sake of clarity, HO-SVD is briefly introduced as follows.

A higher order tensor is usually defined as $A \in R_{i_1 \times \ldots \times i_N}$, where $N$ is the order of $A$, and $1 \leq i_n \leq I_n, 1 \leq n \leq N$. In tensor terminology, the column vectors of a 2-order tensor (matrix) are referred to as mode-1 vectors and row vectors as mode-2 vectors. The mode-n vectors of an N-order tensor $A$ are defined as the $I_n$-dimensional vectors obtained from $A$ by varying the index $i_n$ and keeping the other indices fixed. In addition, a tensor can be expressed in matrix form, which is called matrix unfolding (refer to [3] for details).

Furthermore, the mode-n product, $\times_n$, of a tensor $A \in R_{i_1 \times \ldots \times i_n \times \ldots \times I_N}$ by a matrix $U \in R_{I_n \times i_n}$ along the n-th dimension is defined as,

$$ (A \times_n U)_{i_1,\ldots,i_{n-1},i_n,i_{n+1},\ldots,i_N} = \sum_{i_n} a_{i_1,\ldots,i_{n-1},i_n,i_{n+1},\ldots,i_N} u_{j_n,i_n}. $$

In practice, mode-n multiplication is implemented first by matrix unfolding the tensor $A$ along the given mode-n to generate its mode-n matrix form $A_{(n)}$, and then performing the matrix multiplication as follows,

$$ B_{(n)} = U A_{(n)}. \quad (1) $$

After that, the resulting matrix $B_{(n)}$ is folded back to the tensor form, i.e. $A \times_n U = \text{fold}_n(U \text{unfold}_n(A))$. In terms of mode-n multiplication, Higher Order SVD of a tensor $A$ can be expressed as,

$$ A = S \times_1 U^{(1)} \times_2 \ldots \times_N U^{(N)}, \quad (2) $$

where, $U^{(n)}$ is a unitary matrix of size $I_n \times I_n$, which contains mode-n singular vectors. Instead of being pseudo-diagonal (nonzero elements only occur when the indices $i_1 = \ldots = i_N$), the tensor $S$ (called core tensor) is all-orthogonal, that is, the two subtensors $S_{i_a=a}$ and $S_{i_b=b}$ are orthogonal for all possible values of $n, a$ and $b$ subject to $a = b$. In addition, the Frobenius-norms $\sigma_{(n)}^{i} = \|S_{i_n=i}\|$ are mode-n singular values of $A$ and are in decreasing order, $\sigma_{(1)}^{n} \geq \ldots \geq \sigma_{(n)}^{n} \geq 0$, which correspond to mode-n singular vectors $u_{(n)}^{i} \subset U^{(n)}, i = 1 \ldots I_n$ respectively. The numerical procedure of HO-SVD can be simply described as, unfold$_n(A) = U^{(n)} \Sigma^{(n)} V^{(n)^T}, n = 1 \ldots N$, where, $\Sigma^{(n)} = \text{diag}(\sigma_{(1)}^{n}, \ldots, \sigma_{(n)}^{n})$.

3. nD-PCA

When the PCA technique is applied to n-dimensional dataset, the first encountered numerical problem is to construct the covariance of the tensor. This will incur complicated tensor computations, such as tensor product, generalized transpose and Hermitian symmetrization. In order to avoid them, let’s first re-consider the covariance matrix in the procedure of PCA. Indeed, the covariance matrix can be rewritten in matrix form as follows,

$$ Cov = \frac{1}{M} ((X_1 - \bar{X}), \ldots, (X_M - \bar{X})) (X_M - \bar{X})^{T} = DD^{T}, $$

where, $\bar{X}$ denotes the mean of the training samples, and $D$ denotes the difference matrix of samples. Taking the eigenvalue decomposition of $Cov$ is equivalent to the multiplication of the SVD of $D$ as follows, $Cov = U \Sigma U^{T} = U \Sigma V^{T} V U^{T}$. It is clear that one can construct the difference matrix $D$ rather than the symmetric covariance matrix $Cov$, and apply SVD to $D$ to get the principal orthogonal vectors.

In a similar manner, we can also construct a difference tensor instead of the covariance tensor as follows,

$$ D = ((X_1 - \bar{X}), \ldots, (X_M - \bar{X})) \quad (3) $$

where, $X_i \in R_{I_1 \times \ldots \times I_1 \times \ldots \times I_N}, D \in R_{I_1 \times \ldots \times M I_1 \times \ldots \times I_N}$, i.e. N-order tensors $(X_n - \bar{X}), n = 1 \ldots N$ are stacked along the n-th dimension in the tensor $D$. Furthermore, applying HO-SVD of Eq.(2) to $D$ will generate mode-n singular vectors contained in $U^{(n)}, n = 1 \ldots N$. According to the mode-n singular values, one can determine the desired principal orthogonal vectors for each mode of the tensor $D$ respectively except mode-i singular vectors in Eq.(3). The projection of sample $X$ on mode-n principal vectors $U_{(n)}^{k}$ is expressed as,

$$ Y_n = (X - \bar{X}) \times_n U_{(n)}^{k} \quad (4) $$

It can be seen that the projection $Y_n$ is still an N-order tensor. Projecting $X$ on a principal vector $u_{(n)}^{i}$ of $U_{(n)}^{k}$ will yield $Y_{n,i} = (X - \bar{X}) \times_n u_{(n)}^{i}, i \leq k$, where, where $Y_{n,i} \subset Y_n$ and $Y_{n,i} \in R_{I_1 \times \ldots \times I_{n-i} \times I_{n+1} \ldots \times I_N}$. The projection $Y_{n,i}$ is a $(N-1)$-order tensor. The novel compact form of sample $X$ is described as, $\bar{X} = Y_n \times_n U_{(n)}^{k} + \bar{X}$. For classification, the Frobenius-norms between two mode-n principal component tensors, $Y_n^i$ and $Y_n^j (i \neq j)$, is adopted as follows,

$$ d(Y_n^i, Y_n^j) = \|Y_n^i - Y_n^j\|_F \quad (5) $$

4. Experiments and Analysis

The proposed nD-PCA approach was performed on a 3D range database of human faces used for the Face Recognition Grand Challenge [4]. In order to establish an analogy with a 3D volume dataset or higher dimensional solid dataset, the 3D range dataset was first mapped to a 3D array and the intensity of the corresponding pixels.
in the original still face image was regarded as the voxel value of this 3D array. For the sake of memory size, the reconstructed volume dataset was then re-sampled to the size of $180 \times 180 \times 90$. The original (and re-sampled) 3D range data models and still images of one sample are shown in Fig.1.

Fig.1. The original 3D model and 2D still image of a sample are shown in (a) and (b) while its resampled results are shown in (c) and (d).

**Experiment 1.** The objective of the first experiment is to test the rank of the singular values. In our gallery, eight samples of each person were available for training. Their mean-offset tensors were aligned together along the second index ($x$ axis) to construct a difference tensor $D \in R^{180 \times 1440 \times 90}$. We applied HO-SVD of Eq.(2) to $D$ to get the mode-1 and mode-3 singular values of $D$, which are depicted in Fig.2a. One can note that the numbers of mode-1 and mode-3 singular values are different, and they are equal to the dimensions of indices 1 and 3 of $D$ respectively (i.e. 180 for mode-1 and 90 for mode-3). This is a particular property of higher order tensors, namely the $N$-order tensor $A$ can have $N$ different mode-n rank but they all are less than the rank of $A$, $\text{rank}_n(A) \leq \text{rank}(A)$. Furthermore, the corresponding mode-n singular vectors constitutes orthonormal basis which can span independent mode-n orthogonal spaces respectively. Therefore, we can project a sample to an arbitrary mode-n orthogonal space accordingly. In addition, one can also note that the magnitude of the singular values declines very quickly. This indicates that the energy of a sample is only concentrated on a small number of singular vectors as expected.

![Fig.2a](image1)

![Fig.2b](image2)

Fig.2. (a) The singular values in decreasing order. (b) Comparison of the reconstruction through mode-1, mode-3 and mode1+mode3 principal subspaces respectively. One can note that there is no substantial difference between these three curves.

**Experiment 2.** The second experiment is to test the quality of the reconstructed sample. Within our 3D volume dataset, we had mode-1 and mode-3 singular vectors, which could span two independent orthogonal spaces respectively. Through the selection of the first $k$ principal vectors from mode-1 and mode-3 singular vectors, we can obtain two independent principal subspaces. Our objective is to test which one gives the best reconstruction quality.

We designed a series of tests for this purpose. The reconstructed sample using Eq.(4) was performed on mode-1, mode-3 and mode1+mode3 principal subspaces respectively (i.e. in this last case, a sample was first projected to mode-1, and then to mode-3 principal subspaces, $Y_{13} = (X - \bar{X}) \times_1 U_k^{(1)T} \times_3 U_k^{(3)T}$) with a varying number of principal components $k$. The residual errors of reconstruction are calculated using Eq.(5), and are plotted in Fig.2b. Since the dimensions of $U^{(1)}$ and $U^{(3)}$ are different, the ranges of the corresponding number of principal components $k$ are also different. However, $k$ must be less than the dimension of the corresponding orthogonal matrix $U^{(1)}$ or $U^{(3)}$. As a result of the differing dimensions, the residual error of reconstruction in mode-3 principal subspace converges to zero faster than in mode-1 or mode-1+mode-3 principal subspaces. However, if the curve of mode-3 (solid curve) is quantified to the same length of the corresponding orthogonal matrix $U^{(1)}$ or $U^{(3)}$, it can be noted that the curve of mode-1+mode-3 is similar to the curve of mode-1 (dashed line) in Fig.2b. This indicates that the combination of the mode-1 and the mode-3 principal subspaces cannot further improve the reconstruction quality.

![Fig.3](image3)

**Discussion.** In the above introduction of 1D-, 2D- and nD-PCA, the over-compressed problem was addressed repeatedly. [7] gave a comparison of the reconstruction results between the 1D-PCA case and the 2D-PCA case, which is reproduced in Fig.3 for the sake of completeness. It can be noted that the small number of

![Fig.4](image4)

![Fig.5a](image5)
principal components of the 2D-PCA can perform well compared with the large number of principal components of the 1D-PCA. Moreover, we compared the reconstructed results of the 2D-PCA and nD-PCA with a varying number of principal components $k$ (i.e. the reconstruction of the volume dataset for nD-PCA while the reconstruction of the corresponding 2D image for 2D-PCA). The training set is the same as in the first experiment. The residual errors of reconstruction are normalized to the range of $[0, 1]$, and are plotted in Fig.4. One can note that 2D-PCA performs a little better than nD-PCA in the case of a small number of principal components. In our opinion, there is no visible difference in the reconstruction quality between 2D-PCA and nD-PCA, i.e. like 2D-PCA, nD-PCA does not cause the over-compressed problem. This is because the reconstructed 3D volume dataset is a sparse 3D array (i.e. only the voxel values on the face surface are not equal to zero but all the others are equal to zero), it is therefore more sensitive to computational errors compared to a 2D still image. If the 3D volume dataset were solid, e.g. CT or MRI volume datasets, this difference between the two curves of Fig.4 would not noticeably appear. However, as opposed to 1D-PCA, the 2D- and nD-PCA can both alleviate the over-compression problem.

5. Conclusions

In this paper, we proposed a novel nD-PCA approach, which extends the PCA technique to a higher dimensional dataset. It is indeed a generalized form of the PCA approach for an arbitrary dimensional dataset. This was made possible through the use of tensors and Higher Order Singular Value Decomposition. We tested the proposed approach on 3D volume datasets to test the singular value distribution, and the approximating error. The results indicated that the nD-PCA performed as well as we desired. However, the computational complexity is still a challenging problem in the case of nD-PCA approach. Due to the higher dimensional array, the size of each dataset is so large that all the same kind of samples in the training set cannot be dealt with at the same time. An intuitive idea is to partition a huge multidimensional array to many small blocks, and then to deal with them individually. How to implement this blocking process needs to be considered in future work. In addition, due to the large datasets used, the computation is usually time-consuming, and ways on how to improve the computational speed needs also to be further considered.

References