Tensor Reduction Error Analysis – Applications to Video Compression and Classification

Chris Ding, Heng Huang, and Dijun Luo
Computer Science and Engineering Department
University of Texas, Arlington, Texas, 76019
{chqding, heng, dluo}@uta.edu

Abstract

Tensor based dimensionality reduction has recently been extensively studied for computer vision applications. To our knowledge, however, there exist no rigorous error analysis on these methods. Here we provide the first error analysis of these methods and provide error bound results similar to Eckart-Young Theorem which plays critical role in the development and application of singular value decomposition (SVD). Beside performance guarantee, these error bounds are useful for subspace size determination according to the required video/image reconstruction error. Furthermore, video surveillance/retrieval, 3D/4D medical image analysis, and other computer vision applications require particular reduction in spatio-temporal space, but not along data index dimension. This motivates a D-1 tensor reduction. Standard method such as high order SVD (HOSVD) compress data in all index dimensions and thus can not perform the classification and pattern recognition tasks. We provide algorithm and error bound analysis of the D-1 factorization for spatio-temporal data dimensionality. Experiments on video sequences demonstrate our approach outperforms the previous dimensionality deduction methods for spatio-temporal data.

1. Introduction

Tensor based dimensionality reduction has recently been extensively studied for computer vision applications. Video surveillance/retrieval, 3D/4D medical image analysis, and other computer vision applications require particular dimensionality reduction in spatio-temporal space. At the beginning, standard Principal component analysis (PCA) was used to reduce feature dimensionality. *E.g.* Sirovich and kirby used PCA for human facial images [8]; Turk and Pentland [9] proposed the well-known PCA based eigenface method for face recognition. PCA works well for vector dimensionality reduction, but it is not natural to apply PCA into two dimensional images. In computer vision area, there are several tensor based methods have been proposed. Shashua and Levine [10] employed rank-1 decomposition [13] to represent images; Yang *et al.* [15] proposed a two dimensional PCA (2DPCA). Ye *et al.* [16] proposed a method called Generalized Low Rank Approximation of Matrices (GLRAM) to project the original data onto a two dimensional space. Ding and Ye proposed a non-iterative algorithm called two dimensional singular value decomposition (2DSVD) [3]. There are several other 3D tensor factorization methods and they have been proved to be equivalent to 2DSVD and GLRAM in [7]. For higher dimensional tensor, Vasilescu and Terzopoulos [14] used high order singular value decomposition (HOSVD) [2].

Although many tensor factorization methods have been proposed, to our knowledge, there exist no rigorous error analysis on these methods. In this paper, we provide the first error analysis of these methods and provide error bound results similar to Eckart-Young Theorem which plays critical role in the development and application of SVD. Beside performance guarantee, these error bounds are useful for subspace size determination according to the required video/image reconstruction error. In the real world case, we usually have an expectation on the feature reduction or video/image reconstruction errors and want to balance the errors with subspace size which is related to time and space complexity. Using our error bound, people can easily decide the subspace size without running any program. Furthermore, in order for classification and clustering usage, video surveillance, retrieval, and other computer vision applications always require particular reduction in spatio-temporal space, but not along data index dimension. We propose a D-1 tensor factorization approach that not only reduce the data dimensionality, but also work well for classification and clustering. Standard methods such as HOSVD compress data in all index dimensions and thus can not perform the classification and pattern recognition tasks. If people ignore the compression along data index dimension, HOSVD can be used for D-1 reduction, but with basis not optimized for this purpose. The contributions this paper are summarized as follows:

1) We provide the first error bound analysis of tensor factorization methods with theoretical proof.

2) A novel D-1 tensor factorization method is proposed for spatio-temporal data dimensionality reduction without compression on data index dimension.

3) Experiments on video sequences demonstrate our approach outperforms the previous dimensionality reduction methods on spatio-temporal data classification, retrieval, and recognition.

2. Error analysis of tensor factorization

Although several studies on tensor factorization appeared, to our knowledge, there exist no known error analysis on tensor reduction. In this proposal, we propose to develop a systematic framework for tensor error analysis.

2.1. 2D tensor: PCA

Consider a set of input data vectors $X = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ which can be also viewed as a 2D tensor $X = \{X_{ij}\}_{i=1}^{m}\,_{j=1}^{n}$. PCA is the most widely used dimension reduction method by finding the optimal subspace defined (spanned) by the principal directions $U = (\mathbf{u}_1, \cdots, \mathbf{u}_k) \in \Re^{m\times k}$. The projected data point in the new subspace are $V = (\mathbf{v}_1, \cdots, \mathbf{v}_n) \in \Re^{k\times n}$. PCA finds U and V by minimizing

$$\min_{U,V} J_{PCA} = ||X - UV||_F^2 \tag{1}$$

In PCA or SVD, the Eckart-Young Theorem plays a fundamental role. Eckart-Young Theorem [4] states the optimization problem has PCA/SVD as its global solution and and the optimal (minimum) value is the sum of eigenvalues.

$$J_{PCA}^{\text{opt}} = \sum_{m=k+1}^{\min(p,n)} \lambda_m. \tag{2}$$

where λ_m are eigenvalues of the covariance matrix XX^T. Our main results of this paper is to extend this theorem to tensor factorizations.

2.2. 3D tensor

The input data is a 3D tensor: $X = \{X_{ijk}\}_{i=1}^{n_1}\,_{j=1}^{n_2}\,_{k=1}^{n_3}$. The rank-1 and HOSVD factorizations, treating every index uniformly, is

$$\min_{U,V,W,M} J_1 = ||X - U \otimes_1 V \otimes_2 W \otimes_3 M||^2 \tag{3}$$

where U, V, W are 2d matrices and M is a 3D tensor. Using explicit index, $J_1 = \sum_{ijk}\Big(X_{ijk} - \sum_{pqr} U_{ip}V_{jq}W_{kr}M_{pqr}\Big)^2$ For HOSVD, $M \in \Re^{k_1\times k_2\times k_3}$. For rank-1 decomposition, M is diagonal: $M_{pqr} = m_p$ if $p = q = r$, $M_{pqr} = 0$ otherwise. In many cases, we require that W, U, V are orthogonal: $U^TU = I, V^TV = I,\ W^TW = I$.

We present main results of this paper on extending Eckart-Young type theorems to J_1 in Eq.(3),

Theorem 1 *The factorization of J_1 in Eq.(3), has the upper and lower error bounds*

$$\sum_{m=k_1+1}^{n_1} \lambda_m^F \le J_1^{opt} \le \sum_{m=k_1+1}^{n_1} \lambda_m^F + \sum_{m=k_2+1}^{n_2} \lambda_m^G + \sum_{m=k_3+1}^{n_3} \lambda_m^H.$$

where $(\lambda_1^F, \cdots, \lambda_{n_1}^F)$ are eigenvalues of matrix F, $(\lambda_1^G, \cdots, \lambda_{n_2}^G)$ are eigenvalues of matrix G, $(\lambda_1^H, \cdots, \lambda_{n_3}^H)$ are eigenvalues of matrix H. F, G, H are appropriate covariance matrices defined below in Eqs.(14,15,16).

Remark **1** *In the Theorem for 3D tensor X_{ijk}, F deals with index i, G deals with index j, H deals with index k. The theorem holds for any index correspondence. For example, we can let F deals with j, G deals with k, H deals with i.*

We outline the proof of this theorem.

A) 3-step up-bounding strategy.

Using the following inequality

$$\begin{aligned} |a-b| &= |a - a_1 + a_1 - a_2 + a_2 - b| \\ &\le |a - a_1| + |a_1 - a_2| + |a_2 - b|, \end{aligned}$$

we obtain

$$\begin{aligned} & ||Y - U \otimes_1 V \otimes_2 W \otimes_3 M|| && (4)\\ \le\ & ||Y - U \otimes_1 \bar{M}|| && (5)\\ +\ & ||U \otimes_1 \bar{M} - U \otimes_1 V \otimes_2 \tilde{M}|| && (6)\\ +\ & ||U \otimes_1 V \otimes_2 \tilde{M} - U \otimes_1 V \otimes_2 W \otimes_3 M|| && (7)\\ =\ & ||Y - U \otimes_1 \bar{M}|| && (8)\\ +\ & ||\bar{M} - V \otimes_2 \tilde{M}|| && (9)\\ +\ & ||\tilde{M} - W \otimes_3 M||. && (10) \end{aligned}$$

From Eq.(6) to Eq.(9), U drops out because $U^TU = I$. From Eq.(7) to Eq.(10), U, V both drop out because $V^TV = I$.

The above inequality suggests a 3-step optimization procedure to obtain a good feasible solution for J_1 of Eq.(3).

Step-1:

$$\min_{\substack{U \in \Re^{n_1\times k_1}\\ \bar{M} \in \Re^{k_1\times n_2\times n_3}}} J_u = ||Y - U \otimes_1 \bar{M}|| \tag{11}$$

Step-2: we fix $\bar{M}$ to the values obtained Step-1 and minimize

$$\min_{\substack{V \in \Re^{n_2\times k_2}\\ \tilde{M} \in \Re^{k_1\times k_2\times n_3}\\ \bar{M}\text{fixed}}} ||\bar{M} - V \otimes_2 \tilde{M}|| \tag{12}$$

Step-3: we fix $\bar{M}$ to the values obtained in Step-2, and minimize

$$\min_{\substack{W \in \Re^{n_3 \times k_3} \\ M \in \Re^{k_1 \times k_2 \times k_3} \\ \tilde{M} \text{fixed}}} ||\tilde{M} - W \otimes_3 M|| \tag{13}$$

The benefits of this 3-step approach is that the optimizations in step-2 and step-3 can be solved in exactly the same way as in step-1. In fact, we have

Theorem 2 *The tensor reduction of Eq.(11) has the following global solution.*

$$\bar{M}_{i_1 jk} = \sum_i (U^T)_{i_1 i} X_{ijk}, \quad U = (\mathbf{u}_1, \cdots, \mathbf{u}_k),$$

where $\mathbf{u}_k$ is the eigenvector of the matrix F,

$$F_{ii'} = \sum_{jk} X_{ijk} X_{i'jk} \tag{14}$$

Proof is skipped due to space limit.

Applying Theorem 2 to the Step-2 and Step-3 optimizations, the solution are $V = (\mathbf{v}_1, \cdots, \mathbf{v}_k)$, $W = (\mathbf{w}_1, \cdots, \mathbf{w}_k)$, as the eigenvectors of the covariance matrices G, H:

$$G_{jj'} = \sum_{ii'k} X_{ijk} (UU^T)_{ii'} X_{i'j'k} \tag{15}$$

$$H_{kk'} = \sum_{ii'jj'} X_{ijk} (UU^T)_{ii'} X_{i'j'k'} (VV^T)_{jj'}, \tag{16}$$

and

$$M_{i_1 j_1 k_1} = \sum_{ijk} U_{ii_1} V_{jj_1} W_{kk_1} X_{ijk}. \tag{17}$$

B) Now we can prove rigorously (proof is skipped due to space limit.)

Proposition 3 *Using the solutions (U^*, V^*, W^*) provided by the 3-step procedure, the objective function value*

$$J_1(U^*, V^*, W^*) = \sum_{m=k_1+1}^{n_1} \lambda_m^F + \sum_{m=k_2+1}^{n_2} \lambda_m^G + \sum_{m=k_3+1}^{n_3} \lambda_m^H. \tag{18}$$

C) Now we prove Theorem 1. By counting the degrees of freedom, we have the following

$$J_u \leq J_1^{opt} \leq J_1(U^*, V^*, W^*) \equiv J_{\text{3step}} \tag{19}$$

where J_u from Eq.(11). The left inequality is due to the fact that the J_u optimization is over a larger space than the the J_1 optimization The second inequality holds because $J_1(U^*, V^*, W^*)$ is a specific feasible solution to the J_1 optimization, thus provides a upper bound. From this inequality, we obtain the main inequality of Theorem 1. □

Remark **2** *(U^*, V^*, W^*) can be used as a good initial (U, V, W), since the upper bound is tight.*

2.3. 4D Tensor

Suppose the input data is $Y = \{Y_{ijk\ell}\}_{i=1}^{n_1}{}_{j=1}^{n_2}{}_{k=1}^{n_3}{}_{l=1}^{n_4}$. A symmetric factorization (such as HOSVD factorizations) would factorize Y into $Y \simeq U \otimes_1 V \otimes_2 W \otimes_3 S \otimes_4 M$. i.e.,

$$\min_{U,V,W,S,M} J_4 = ||Y - U \otimes_1 V \otimes_2 W \otimes_3 S \otimes_4 M||^2 \tag{20}$$

Error analysis of Theorem 1 can generalize directly to 4D tensors in a obvious way.

3. D-1 orthogonal tensor decomposition

We consider factorization of D-dimensional tensors using D-1 orthogonal subspaces. In this case, only D-1 index dimensions are compressed, but the data dimension are not compressed. This approach has been used before implicitly. Here we study this approach formally and systematically. We give computational algorithms and provide error analysis.

3.1. 2D tensor

We first motivate this approach using PCA example. We can write the PCA objective function as

$$J_{PCA} = ||X - UV||_F^2 = \sum_{i=1}^{n} ||\mathbf{x}_i - U\mathbf{v}_i||^2. \tag{21}$$

See Eq.(1). Thus for 2D tensor, the reduction is to a D-1=1-dimensional subspace U.

3.2. 3D tensor

The input data 3D tensor $X = \{X_{ijk}\}_{i=1}^{n_1}{}_{j=1}^{n_2}{}_{k=1}^{n_3}$ can be viewed as $X = \{X_1, \cdots, X_{n_3}\}$ where each X_i is a 2d matrix (an image) of size $n_1 \times n_2$. Therefore, instead of treating every index equally as in J_1 of Eq.(3), we leave the data index uncompressed and optimize

$$\min_{U,V,M} J_2 = ||X - U \otimes_1 V \otimes_2 M||^2 = \sum_{\ell=1}^{n_3} ||X_\ell - U \otimes_1 V \otimes_2 M_\ell||^2 \tag{22}$$

where we let $M = \{M_1, \cdots, M_{n_3}\}$. By definition,

$$[U \otimes_1 V \otimes_2 M_\ell]_{ij} = \sum_{p,q} U_{ip} V_{jq} M_{pq\ell}$$

$$= \sum_{p,q} U_{ip} (M_\ell)_{pq} (V^T)_{qj} = (U M_\ell V^T)_{ij}.$$

Thus we can write D-1 factorization for a 3D tensor as

$$\min_{U,V,M_\ell} J_3 = \sum_{\ell=1}^{n_3} ||X_\ell - U M_\ell V^T||^2 \tag{23}$$

which is identical to GLRAM/2DSVD [3, 16].

Thus D-1 factorization reduces to known factorizations for 2D and 3D tensors. The 2DPCA of Yang *et al.* [15] is a special case of Eq.(23) by setting $U = I$ (*i.e.* ignoring U and increasing the size of M_ℓ from $k_1 \times k_2$ to $r \times k_2$).

3.3. 4D tensors

The input data 4D tensor $Y = \{Y_{ijk\ell}\}_{i=1}^{n_1}{}_{j=1}^{n_2}{}_{k=1}^{n_3}{}_{l=1}^{n_4}$ can be viewed as $Y = \{Y_1, \cdots, Y_{n_4}\}$ where each Y_i is a 3D tensor (a cube, or a video consisting a set of 2D images). In contrast to J_4 of Eq.(20), we here compress 3 dimensions of each 3D tensor Y_i, but not on the data index dimension:

$$\min_{U,V,W,M} J_5 = \sum_{\ell=1}^{n_4} ||Y_\ell - U \otimes_1 V \otimes_2 W \otimes_3 M_\ell||^2 \quad (24)$$

We have set $M = \{M_1, \cdots, M_{n_4}\}$. Computational algorithm will be presented in §3.7.

3.4. Robust D-1 tensor factorization

A robust version of D-1 tensor factorization also exists [6] using R_1 norm and robust covariance matrices.

3.5. Two reasons for D-1 tensor factorization

There are two main reasons why D-1 Tensor Factorization is preferable: (1) classification, i.e., object recognition, (2) clustering, i.e., automatic pattern discovery.

Classification

Consider 3D tensor with input data: $X = \{X_1, \cdots, X_n\}$ where each X_i is a 2d matrix (an image). Suppose we obtained D-1 factorization solutions: $U, V, \{M_\ell\}$. In image retrieval, recognition, classification tasks, we are given a query image, and wish to check the database to find the image closest to the query image. This involves the distance. Given two images X_i, X_j, their distance in the tensor subspace can be efficiently computed as

$$||X_i - X_j||^2 = ||LM_iR^T - LM_iR^T||^2 = ||M_i - M_j||^2$$

Consider 4d tensor with input data: $Y = \{Y_\ell\}_{\ell=1}^{\ell=n}$ where each Y_i is a 3D tensor (a video of fixed number of frames). Suppose we obtained D-1 factorization solutions: 2d matrices U, V, W and the 3D tensor $\{M_\ell\}$. The distances between two videos Y_i, Y_j are efficiently computed as

$$||Y_i - Y_j||^2 = ||UVW(M_i - M_j)||^2 = ||M_i - M_j||^2$$

Tensor clustering

Given a set of 1d tensors (vectors) $x_1, x_2, \cdots, x_n$, we can do K-means clustering

$$\min_{\{c_k\}} \sum_{\ell=1}^{n} \min_{1\le k\le K} ||x_\ell - c_k||^2 = \sum_{k=1}^{K} \sum_{i\in c_k} ||x_\ell - c_k||^2 \quad (25)$$

where c_k is the centroid vector of cluster c_k. This formalism can be extended to generic tensors. Given an d dimensional tensor, or, equivalently a set of (D-1)-dimensional tensors $X^1, X^2, \cdots, X^n$, the K-means tensor clustering minimizes

$$\min_{\{C^{(k)}\}} \sum_{\ell=1}^{n} \min_{1\le k\le K} ||X^{(\ell)} - C^{(k)}||^2 = \sum_{k=1}^{K} \sum_{i\in C_k} ||X^{(\ell)} - C^{(k)}||^2 \quad (26)$$

where C_k is the centroid tensor of cluster C_k. Now suppose we carried out a D-1 tensor factorization on X into $U, V, W, \cdots$, and $\{M_\ell\}$. Using the distance relationship, the tensor clustering can be done entirely in $\{M_\ell\}$:

$$\min_{\{C^{(k)}\}} \sum_{\ell=1}^{n} \min_{1\le k\le K} ||M^{(\ell)} - C^{(k)}||^2 = \sum_{k=1}^{K} \sum_{i\in C_k} ||M^{(\ell)} - C^{(k)}||^2 \quad (27)$$

where C_k is the centroid tensor of cluster C_k. These clustering formulations show the usefulness of D-1 tensor factorization.

3.6. Error analysis of D-1 factorization

Error analysis in §2 can be directly extended to D-1 factorization. We have

Theorem 4 *The factorization of J_5 in Eq.(24) has the upper and lower error bounds as shown in Theorem 1 with F, G, H are appropriate covariance matrices defined below in Eq.(28).*

$$F_{ii'} = \sum_\ell \sum_{jk} Y^{(\ell)}_{ijk} Y^{(\ell)}_{i'jk}, \quad (28)$$

$$G_{jj'} = \sum_\ell \sum_{ii'k} X^{(\ell)}_{ijk} (UU^T)_{ii'} X^{(\ell)}_{i'j'k}, \quad (29)$$

$$H_{kk'} = \sum_\ell \sum_{ii'jj'} X^{(\ell)}_{ijk} (UU^T)_{ii'} X^{(\ell)}_{i'j'k'} (VV^T)_{jj'}. \quad (30)$$

Theorems 1 and 4 holds for arbitrary number of factors (*e.g.* GLRAM/2DSVD for 2D tensors). They are the generalization of Eckart-Young theorem to tensors.

3.7. Using error bounds to determine reduction parameters

Suppose in Theorem 1, with initial k_1^0, k_2^0, k_3^0, matrix F, G, H are formed and eigenvalues are computed. We can used the upper bound to estimate compression error at any k_1, k_2, k_3:

$$J(k_1, k_2, k_3) = \sum_{m=k_1+1}^{n_1} \lambda_m^F + \sum_{m=k_2+1}^{n_2} \lambda_m^G + \sum_{m=k_3+1}^{n_3} \lambda_m^H.$$

This relation is exact when $k_1 = k_1^0, k_2 = k_2^0, k_3 = k_3^0$, For other k_1, k_2, k_3 values: this is a good approximation. We can use this for choosing parameters k_1, k_2, k_3. Given a pre-specified error tolerance on reconstruction error: δ. Simply choose k_1, k_2, k_3 such that $J(k_1, k_2, k_3) \le \delta$.

3.8. Algorithm for D-1 factorizations

We derive the algorithm for 4D tensor D-1 Factorization J_5 in Eq.(24).

First, we initialize U,V, W using Theorem 4, where U^0 is given by the eigenvectors of Eq.(28) and V^0 is given by the eigenvectors of Eq.(29). Second, we iterate to compute newer W, U, V as eigenvectors of H, F, G as

$$H_{kk'} = \sum_{\ell} \sum_{ii'jj'} X_{ijk}^{(\ell)} X_{i'j'k'}^{(\ell)} (UU^T)_{ii'} (VV^T)_{jj'}$$

$$F_{ii'} = \sum_{\ell} \sum_{jj'kk'} X_{ijk}^{(\ell)} X_{i'j'k'}^{(\ell)} (VV^T)_{jj'} (WW^T)_{kk'}$$

$$G_{jj'} = \sum_{\ell} \sum_{ii'kk'} X_{ijk}^{(\ell)} X_{i'j'k'}^{(\ell)} (UU^T)_{ii'} (WW^T)_{kk'}$$

3.9. Time and space complexities

In Table 1, we summarize the storage and matrix sizes for which we need to compute D-1 tensor factorization. For N_l image sequences which include N_1 images of size $N_2 \times N_3$, the number of principle components is selected on three directions as K_1, K_2, K_3, respectively. K is the number of principle components in PCA.

method	storage
PCA	$N_1 N_2 N_3 K + N_l K$
D-1 factorization	$N_1 K_1 + N_2 K_2 + N_3 K_3 + N_l K_1 K_2 K_3$

Table 1. Storage comparison for $N_l \times N_1$ images of size $N_2 \times N_3$.

3.10. Relation to HOSVD

In principle, we can compute HOSVD and use U, V (ignoring W) for clustering and classification tasks. But this is non-optimal, due to the compression of data dimension. Although Tucker-2 decompositions [13] mentioned 3 possible 2-index decompositions for 3D tensor, the significance leaving data dimension uncompressed is first recognized in [3, 16].

4. Experimental results

In this section, we experimentally evaluate the performance of our proposed algorithm with respect to the quality of video classification, information retrieval, and face recognition. The images from ORL face database [1] are used to demonstrate how the error bounds can be used to determinate the subspace size. One public video dataset TRECVID 2005 is used for experiments on video classification and information retrieval. The other well known face database (CMU PIE) [11] are used validate to the performance of D-1 tensor factorization method. In this section, $K_i = K_1$, $K_j = K_2$, $K_k = K_3$, and K_1, K_2, K_3, N_i, N_j, N_k are the same parameters in Table 1.

4.1. Upper bound experiment

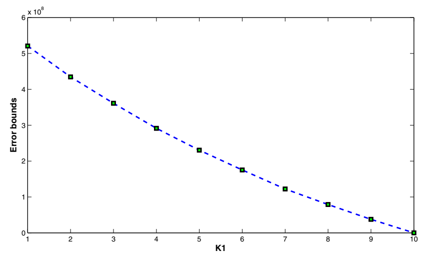
The upper bound in our theorem 1 is astonishing tight. We reconstruct 4D tensor images from AT&T face database [1], YALE face database B [5], and PIE face database [11] with $k_1 = k_2 = 20$, k_3 =the length of image sequence. The error ratio is defined as $(J_{upperbound} - J_1^{opt})/||X||_F^2$. The error ratios of all three datasets are less than 10^{-6}.

4.2. Demonstration of subspace size selection using error bounds

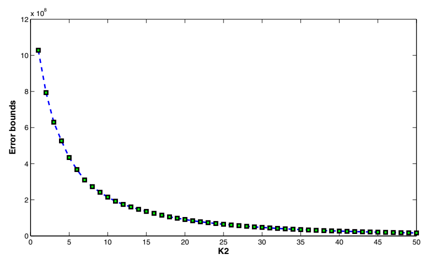
In the ORL database (current AT&T), there are 40 different people and each has ten different images. For some people, the images were taken at different times, varying the lighting, facial expression (open/close eyes, smiling/no-smiling) and facial details (glasses/no glasses). We treat all images as 40 image sequences and each subject has one sequence which has ten different images. Since we only demonstrate how to decide the subspace size using error bounds, the size of dataset is not an issue.

The D-1 tensor factorization is applied into these ten image sequences and the error bounds are calculated using Theorem 1: $B_F = \sum_{m=K_1+1}^{N_i} \lambda_m^F$, $B_G = \sum_{m=K_2+1}^{N_j} \lambda_m^G$, and $B_H = \sum_{m=K_3+1}^{N_k} \lambda_m^H$. The error bound values of B_F, B_G, and B_H are plotted in Fig. 1. When the number of K_1, K_2, and K_3 increase, the error bounds decrease. Since K_1 only can change from 1 to 10, we plot all of them. Because ten images of each sequence are different and have few correlation between each other (*e.g.* continuous movement). The curve in Fig. 1(a) doesn't decrease fast in the first several K_1 values. The image size of ORL face data is 92×112. We show the error bound values of B_G and B_H at K_2 and K_3 ranging from 1 to 50 in Fig. 1(b) and Fig. 1(c). Since the correlations of row-row and column-column in images are very high, these two curves decrease fast during the first several K_2 or K_3 values.

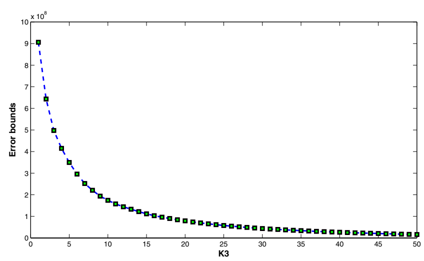
For different datasets, people can easily plot such three figures for error bounds. Using them, the cutoff values of K_1, K_2, and K_3 can be determined. From Theorem 1, we know the sum of error bounds of K_1, K_2, and K_3 is the upper bound of image reconstruction error. Please pay attention to the Remark 1 for Theorem 1. Each index i, j, and k can be used to compute the low bound of reconstruction error.



(a)



(b)



(c)

Figure 1. Error bound values of $B_F = \sum_{m=K_1+1}^{N_i} \lambda_m^F$, $B_G = \sum_{m=K_2+1}^{N_j} \lambda_m^G$, and $B_H = \sum_{m=K_3+1}^{N_k} \lambda_m^H$ using ORL face database.

4.3. Experiments on TRECVID 2005

Using the dataset of TRECVID 2005 [12], the video sequences are constructed as the following steps. We choose the shots in which there are at least 5 sub-shot key frames and select the first 5 key frames to form the sequence. In order for the convenience of evaluation, we ignore the shots which are not labeled in the ground-truth data. Finally we generate 347 video sequences for 10 topics. To evaluate the performance, K-NN classifiers with K=1 are employed both in D-1 tensor factorization method and standard PCA. In PCA, we construct the vector for one video sequence using all the pixels of all the images in the sequence. Since the videos in TRECVID 2005 are multi-labeled, we compare the leave-one-out overall classification precision and recall (commonly also called as sensitivity) with respect to the storage size that can be calculated from Table 1.

The comparison results are shown in Fig. 2 and Fig. 3.

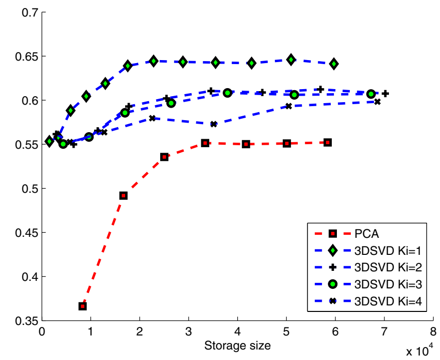


Figure 2. Video classification overall precision comparison between D-1 tensor factorization (K_i from 1 to 4) and PCA.

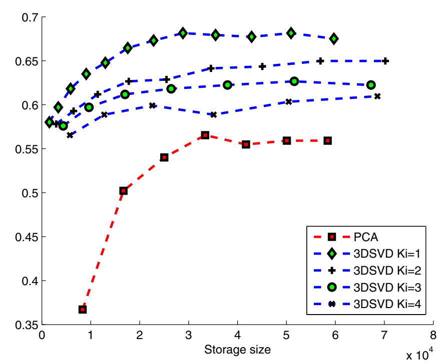


Figure 3. Video classification overall recall comparison between D-1 tensor factorization (K_i from 1 to 4) and PCA.

The values on x-axis are the storage size of both D-1 tensor factorization method and PCA. The values on y-axis in Fig. 2 are classification accuracy and in Fig. 3 are sensitivity. We have the following observations: a) for all K_i from 1 to 4, the D-1 tensor factorization method outperforms PCA on both classification precision and recall; b) because both D-1 tensor factorization and PCA use global features for classification, the classification accuracy is lower than 70% (if more local features are incorporated into global features, the classification can be improved more); c) because the environmental background during these 347 video sequences has a large variation and $K_i = 1$ can filter out most redundant and irrelevant information between images in the sequence, the D-1 tensor factorization method has the best performance under $K_i = 1$.

Using these 347 video sequences, we also perform our method on image retrieval experiment. Fig. 4 plots a case of the retrieval results using both D-1 tensor factorization and PCA. An Euclidean distance is deployed on the compressed feature space as the ranking metric. In Fig. 4, two top rows are retrieval results of PCA and two bottom rows are results of D-1 tensor factorization. In each method, the first image (they are the same query image) is the key frame of query video which belongs to topic 10 (cars related video), the others are the top-9 ranked retrieval videos. PCA retrieves the wrong images from topic 4 and topic 1 at rank 7 and 8. Our D-1 tensor factorization method only gets one error at rank 8. During the retrieval results of our method, two images at rank 6 and 9 are pretty interesting. They don't have car in query video, but they really include different cars and driver inside. Since D-1 tensor factorization method extract the correlation from images in sequence, it is a promising approach for video classification and retrieval.

4.4. PIE database

The CMU Pose, Illumination, and Expression (PIE) database [11] contains 41,368 images of 68 people, each person under 13 different poses, 43 different illumination conditions. We collect the first 30 people with all 13 different poses under 10 randomly selected illumination conditions. We treat the 13 different poses under the same person and illumination condition as a sequence. Thus, every people has 10 image sequences under 10 different illumination conditions. The order of pose in each sequence is fixed. Totally 300 sequences are selected and 10 for each person. We choose $N_j = 40, N_k = 30$ in the face recognition experiment.

For each people, we use 9 image sequences as training data and the other one as testing data. K-NN classifiers with K=1 are employed with 10-fold cross validation. Since each sequence has only one label, we compare the overall face recognition accuracy. Fig. 5 presents the face recognition results. The values on x-axis are storage numbers and on y-axis are classification accuracy. Our D-1 tensor factorization method overwhelmingly outperforms PCA for all K_i from 1 to 4. When PCA uses 2 principle components, the face recognition accuracy of D-1 tensor factorization method is above 80% for all K_i from 1 to 4. As we discussed above, because the environmental background has a large variation, $K_i = 1$ has a better face recognition accuracy.

In the real world case, people's poses are more unpredictable and we are not able to control the capture process of the face video. Thus, we are also interested in the video sequences that are disordered in pose. Based on this truth, we conduct another experiment using the above settings on illustration, but the order of poses in different sequences are random.

We use the same face recognition experimental settings (K-NN classifiers with K=1 plus 10-fold cross validation). Fig. 6 compares the face recognition accuracy between D-1 tensor factorization and PCA. Since the order of poses is random, the distances between sequences within the class is increased and the face recognition accuracy definitely is decreased. PCA has a very low recognition rate. D-1 tensor factorization method still can keep a better face recognition performance through the correlations within image sequence. Our method is more robust to the real world applications compared to PCA.

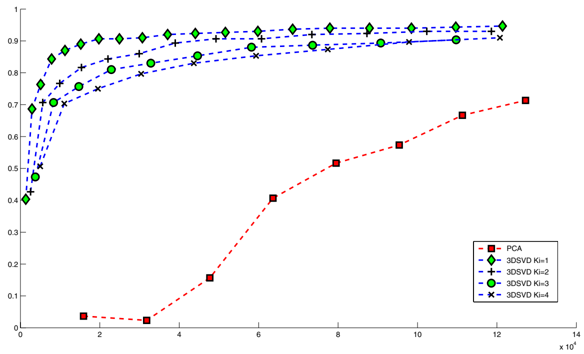


Figure 5. Face recognition accuracy comparison between D-1 tensor factorization (K_i from 1 to 4) and PCA.

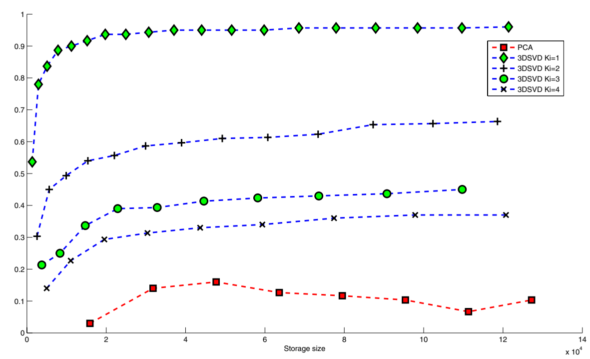


Figure 6. Face recognition accuracy comparison between D-1 tensor factorization (K_i from 1 to 4) and PCA when the order of poses is random.

5. Conclusion

In this paper, we first provide error bounds for various tensor factorizations. This error bound can help users to determine subspace dimensions. The theorems also suggest a way to initialize subspaces. Furthermore, motivated by video classification and recognition, we generalize existing approaches into a D-1 tensor factorization framework and formally analyze its properties. D-1 factorizations are natural for clustering, recognition, and classification. Using the dataset from TRECVID and PIE face databases, we demonstrate our approach has a much better performance than standard PCA on video classification, information retrieval, and face recognition from image sequences.

Figure 4. One retrieval case using both PCA and D-1 tensor factorization. The top two rows are retrieval results of PCA and two bottom rows are results of D-1 tensor factorization. In each method, the first image is the key frame of query video (the first left images on row 1 and 3), the others are the top-9 ranked retrieval videos (starting from the second left images on row 1 and 3).

References

[1] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html. 5

[2] L. de Lathauwer, B. de Moor, and J. Vanderwalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4), 2000. 1

[3] C. Ding and J. Ye. Two-dimensional singular value decomposition (2dsvd) for 2d maps and images. *SIAM Int'l Conf. Data Mining*, pages 32–43, 2005. 1, 3, 5

[4] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:183–187, 1936. 2

[5] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001. 5

[6] H. Huang and C. Ding. Robust tensor facorization using r1 norm. *IEEE Conf. Computer Vision and Pattern Recognition*, 2008. 4

[7] K. Inoue and K. Urahama. Equivalence of non-iterative algorithms for simultaneous low rank approximations of matrices. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:154 – 159, 2006. 1

[8] Kirby and Sirovich. Application of the kl procedure for the characterization of human faces. *IEEE Trans. Pattern Anal. Machine Intell.*, 12:103–108, 1990. 1

[9] T. M. and Pentland. Eigen faces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991. 1

[10] A. Shashua and A. Levin. Linear image coding for regression and classification using the tensor-rank principle. *IEEE Conf. on Computer Vision and Pattern Recognition*, 2001. 1

[11] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database of human faces, 2002. 5, 7

[12] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid, 2006. 6

[13] L. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966. 1, 5

[14] M. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. *European Conf. on Computer Vision*, pages 447–460, 2002. 1

[15] J. Yang, D. Zhang, A. F. Frangi, and J. Yang. Twodimensional pca: A new approach to appearancebased face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1), 2004. 1, 4

[16] J. Ye. Generalized low rank approximations of matrices. *International Conference on Machine Learning*, 2004. 1, 3, 5