Two-Dimensional Singular Value Decomposition (2DSVD) for 2D Maps and Images

Chris Ding* and Jieping Ye†

LBNL-56481. October 3, 2004.

Abstract

For a set of 1D vectors, standard singular value decomposition (SVD) is frequently applied. For a set of 2D objects such as images or weather maps, we form 2DSVD, which computes principal eigenvectors of row-row and column-column covariance matrices, exactly as in the standard SVD. We study optimality properties of 2DSVD as low-rank approximation and show that it provides a framework unifying two recent approaches. Experiments on images and weather maps illustrate the usefulness of 2DSVD.

1 Introduction

Singular value decomposition (SVD)[5, 7] plays the central role in reducing high dimensional data into lower dimensional data which is also called principal component analysis (PCA)[8] in statistics. It often occurs that in the reduced space, coherent patterns can be detected more clearly. Such unsupervised dimension reduction is used in very broad areas such as meteorology[11], image processing[9, 13], and information retrieval[1].

The problem of low rank approximations of matrices has recently received broad attention in areas such as computer vision, information retrieval, and machine learning [1, 2, 3, 12]. It becomes an important tool for extracting correlations and removing noise from data. However, applications of this technique to high-dimensional data, such as images and videos, quickly run up against practical computational limits, mainly due to the high time and space complexities of the SVD computation for large matrices.

In recent years, increasingly more data items come naturally as 2D objects, such the 2D images, 2D weather maps. Currently widely used method for dimension reduction of these 2D data objects is based on SVD. First, 2D objects are converted into 1D vectors and are packed together as a large matrix. For example, each of the 2D maps of $A_i, A_i \in \mathbb{R}^{r\times c}, i = 1, \cdots, n$ is converted to a vector $\mathbf{a}_i$ of length rc. The standard SVD is then applied to the matrix containing all the vectors: $A = (\mathbf{a}_1, \cdots, \mathbf{a}_n)$. In image processing, this is called Eigenfaces[9]. In weather research, this is called Empirical Orthogonal Functions (EOF) [11]. Although the conventional approach is widely used, it does not preserve the 2D nature of these 2D data objects.

Two recent studies made first proposals to capture the 2D nature explicitly in low rank approximation. Yang *et al.* [13] propose to use the principal components of (column-column) covariance matrix for image representation. Ye *et al.* [14, 15] propose to use a LM_iR^T type decomposition for low rank approximation.

In this paper, we propose to construct 2-dimensional singular value decomposition (2DSVD) based on the row-row and column-column covariance matrices. We study various optimality properties of 2DSVD as low-rank approximation. We show that the approach of Yang *et al.* [13] can be casted as a one-sided low-rank approximation with its optimal solution given by 2DSVD. 2DSVD also gives a near-optimal solution for the low rank approximation using LM_iR^T decomposition by Ye [14]. Thus 2DSVD serves as a framework unifying the work of Yang *et al.* [13] and Ye [14].

Together, this new approach captures explicitly the 2D nature and has 3 advantages over conventional SVD-based approach: (1) It deals with much smaller matrices, typically $r \times c$ matrices, instead of $n \times (rc)$ matrix in conventional approach. (2) At the same or better accuracy of reconstruction, the new approach requires substantially smaller memory storage. (3) Some of the operations on these rectangular objects can be done much more efficiently, due to the preservation of the 2D structure.

We note there exists other type of decompositions of high order objects. The recently studied orthogonal tensor decomposition [16, 10], seeks an f-factor trilinear form for decomposition of X into A, B, C: $x_{ijk} = \sum_{\alpha=1}^{f} a_{i\alpha} b_{j\alpha} c_{k\alpha}$ where columns of A, B, C mutually orthogonal within each matrices.

Our approach differs in that we keep explicit the 2D nature of these 2D maps and images. For weather map,

*Lawrence Berkeley National Laboratory, University of California, Berkeley, CA 94720. Email: chqding@lbl.gov

†Department of Computer Science, University of Minnesota, Minneapolis, MN 55455. Email: jieping@cs.umn.edu

the i, j dimensions are longitude and latitude which are of same nature. For 2D images, the i, j dimensions are vertical and horizontal dimensions, which are of the same nature. The k dimension refers to different data objects. (In contrast, in the multi-factor trilinear orthogonal decomposition, the i, j, k dimensions are of different nature, say "temperature", "intensity", "thickness".)

These inherently 2D datasets are very similar to 1D vector datasets, $X = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$, for which the singular value decomposition (SVD) is often applied to obtain the optimal low-rank approximation:

$$X \approx \widetilde{X},\ \widetilde{X} = U_k \Sigma_k V_k^T,\ \Sigma_k = U_k^T X V_k, \tag{1.1}$$

where U_k contains k principal eigenvectors of the covariance matrix[1] XX^T and V contains k principal eigenvectors of the inner-product matrix $X^T X$.

We define 2-dimensional SVD for a set of 2D maps in the same way as SVD is computed for a set of 1D vectors. Define the averaged row-row and column-column covariance matrices,

$$\begin{aligned} F &= \sum_{i=1}^{n} (A_i - \bar{A})(A_i - \bar{A})^T, \\ G &= \sum_{i=1}^{n} (A_i - \bar{A})^T (A_i - \bar{A}). \end{aligned} \tag{1.2}$$

where $\bar{A} = \sum_i A_i / n$.[1] The normalization factor $1/n$ in F, G are ignored since they do not affect the results. F corresponds to XX^T and G corresponds to $X^T X$. Let U_k contains k principal eigenvectors of F and V_s contains s principal eigenvectors of G:

$$F = \sum_{\ell=1}^{r} \lambda_\ell \mathbf{u}_\ell \mathbf{u}_\ell^T,\ U_k \equiv (\mathbf{u}_1, \cdots, \mathbf{u}_k); \tag{1.3}$$

$$G = \sum_{\ell=1}^{c} \zeta_\ell \mathbf{v}_\ell \mathbf{v}_\ell^T,\ V_s \equiv (\mathbf{v}_1, \cdots, \mathbf{v}_s). \tag{1.4}$$

Following Eq.(1.1), we define

$$\tilde{A}_i = U_k M_i V_s^T,\ M_i = U_k^T A_i V_s,\ i = 1, \cdots, n, \tag{1.5}$$

as the extension of SVD to 2D maps. We say $(U_k, V_s, \{M_i\}_{i=1}^n)$ form the 2DSVD of $\{A_i\}_{i=1}^n$. In standard SVD of Eq.(1.1), U_k provides the common subspace basis for 1D vectors to project to. In 2DSVD, U_k, V_s provide the two common subspace bases for 2D maps to (right and left) project to (this will become more clear in §3, §4 §5). Note that $M_i \in \mathbb{R}^{k \times s}$ is not required to be diagonal, whereas in standard SVD, Σ_k is diagonal.

For standard SVD, the eigenvalues of XX^T and $X^T X$ are identical, $\lambda_\ell = \zeta_\ell = \sigma_\ell^2$. The Eckart-Young Theorem[5] states that the residual error

$$\left\| X - \sum_{\ell=1}^{k} \mathbf{u}_\ell \sigma_\ell \mathbf{v}_\ell^T \right\|^2 = \sum_{\ell=k+1}^{r} \sigma_\ell^2. \tag{1.6}$$

We will see that 2DSVD has very similar properties.

Obviously, 2DSVD provides a low rank approximation of the original 2D maps $\{A_i\}$. In the following we provide detailed analysis and show that 2DSVD provides (near) optimal solutions to a number of different types of approximations of $\{A_i\}$.

2 Optimality properties of 2DSVD

Definition. Given a 2D map set $\{A_i\}_{i=1}^n$, $A_i \in \mathbb{R}^{r \times c}$, we define the low rank approximation

$$\begin{aligned} A_i &\approx \tilde{A}_i, \quad \tilde{A}_i = LM_iR^T, \\ L &\in \mathbb{R}^{r \times k},\ R \in \mathbb{R}^{c \times s},\ M_i \in \mathbb{R}^{k \times s}. \end{aligned} \tag{2.7}$$

Here k, s are input parameters for specifying the rank of the approximation. We require L, R have orthonormal columns $L^T L = I_k,\ R^T R = I_s$. A less strict requirement is: columns of L be linearly independent and columns of R be linearly independent. However, given a fixed L, R with these constraints, we can do QR factorization to obtain $L = Q_L \tilde{L}$ and $R = Q_R \tilde{R}$ where Q_L, Q_R are orthogonal. We can write $LM_iR^T = Q_L \tilde{L} M_i \tilde{R} Q_R^T = Q_L \tilde{M}_i Q_R^T$. This is identical to the form of LM_iR^T.

The 2DSVD of Eq.(1.5) is clearly one such approximation. What's the significance of 2DSVD?

(S1) The optimal solution for the low-rank approximation using the *1-sided* decomposition

$$\min_{M_i \in \mathbb{R}^{r \times k}, R \in \mathbb{R}^{c \times k}} J_1(\{M_i\}, R) = \sum_{i=1}^{n} ||A_i - M_i R^T||^2 \tag{2.8}$$

is given by the 2DSVD: $R = V_k,\ M_i = A_i V_k$. This case is equivalent to the situation studied by Yang *et al.*[13].

(S2) The optimal solution for the *1-sided* low-rank approximation

$$\min_{L \in \mathbb{R}^{r \times k}, M_i \in \mathbb{R}^{c \times k}} J_2(L, \{M_i\}) = \sum_{i=1}^{n} ||A_i - LM_i^T||^2 \tag{2.9}$$

is given by the 2DSVD: $L = U_k,\ M_i = A_i^T U_k$.

(S3) The 2DSVD gives a near-optimal solution for the low-rank approximation using the 2-sided decomposition [14]

$$\min_{L \in \mathbb{R}^{r \times k}, R \in \mathbb{R}^{c \times s}, M_i \in \mathbb{R}^{k \times s}} J_3(L, \{M_i\}, R) = \sum_{i=1}^{n} ||A_i - LM_iR^T||^2. \tag{2.10}$$

[1] In general, SVD is applied to any rectangular matrix, while PCA applies SVD on centered data: $X = (\mathbf{x}_1 - \bar{\mathbf{x}}, \cdots, \mathbf{x}_n - \bar{\mathbf{x}})$, $\bar{\mathbf{x}} = \sum_i \mathbf{x}_i / n$. In the rest of this paper, we assume $\bar{A} = 0$ to simplify the equations. For un-centered data, corresponding equations can be recovered by $A_i \rightarrow A_i - \bar{A}$.

When $k = r$, $\min J_3$ reduces to $\min J_1$. When $s = c$, $\min J_3$ reduces to $\min J_2$.

(S4) When $A_i = A_i^T, \forall i$, the 2DSVD gives a near-optimal solution for the symmetric approximation

$$(2.11) \quad \min_{L \in \mathbb{R}^{r\times k}, M_i \in \mathbb{R}^{k \times k}} J_4(L, \{M_i\}) = \sum_{i=1}^{n} ||A_i - L M_i L^T||^2.$$

2DSVD provides a unified framework for rectangular data matrices. Our 2DSVD generalizes the work of Yang *et al.* [13] which is equivalent to (S1), but their feature extraction approach is different from our decomposition approach with the optimization of an objective function. On other hand, the 2DSVD provides a near-optimal solution of the 2D low rank approximation of Ye [14], the symmetric decomposition of J_3 which we believe is key to the low rank approximation of these rectangular data matrices.

We discuss these decompositions in §3, §4, §5.

3 $A_i = M_i R^T$ Decomposition

Theorem 1. The *global* optimal solution for $A_i = M_i R^T$ approximation of J_1 in Eq.(2.8) is given by

$$(3.12) \quad \begin{aligned} R &= V_k, \; M_i = A_i V_k, \\ J_1^{\text{opt}} &= \sum_i ||A_i - A_i V_k V_k^T||^2 = \sum_{j=k+1}^{c} \zeta_j. \end{aligned}$$

Remark. Theorem 1 is very similar to Eckart-Young Theorem of Eq.(1.6) in that the solution is given by the principal eigenvectors of the covariance matrix and the residual is the sum of the eigenvalues of the retained subspace.

Note that this solution is unique[2] up to an arbitrary k-by-k orthogonal matrix Γ: for any given solution $(L, \{M_i\})$, $(L\Gamma, \{M_i \Gamma\})$ is also a solution with the same objective value. When $k = c$, R becomes a full rank orthogonal matrix, i.e., $RR^T = I_c$. In this case, we set $R = I_c$ and $M_i = A_i$.

Proof. Using $||A||^2 = \text{Tr}(A^T A)$, and $\text{Tr}(AB) = \text{Tr}(BA)$, we have

$$\begin{aligned} J_1 &= \sum_{i=1}^{n} \text{Tr}(A_i - M_i R^T)^T (A_i - M_i R^T) \\ &= \text{Tr} \sum_{i=1}^{n} [A_i^T A_i - 2A_i^T M_i R^T + M_i M_i^T] \end{aligned}$$

This is a quadratic function w.r.t. M_i. The minimum occur at where the gradient is zero: $0 = \partial J_1 / \partial M_i = -2A_i R + 2M_i$. Thus $M_i = A_i R$. With this, we have

$$J_1 = \sum_{i=1}^{n} ||A_i||^2 - \text{Tr}[R^T (\sum_{i=1}^{n} A_i^T A_i) R]$$

Now $\min_R J_1$ becomes

$$\max_{R | R^T R = I_k} J_{1a} = \text{Tr}(R^T G R)$$

By a well-known result in algebra, the optimal solution for R is given by $R = (\mathbf{v}_1, \cdots, \mathbf{v}_k)\Gamma$, Γ is an arbitrary k-by-k orthogonal matrix noted earlier. The optimal value is the sum of the large k eigenvalues of G: $J_{1a}^{\text{opt}} = \sum_{j=1}^{k} \zeta_j$. Note that

$$(3.13) \quad \sum_{j=1}^{c} \zeta_j = \text{Tr}(V_c^T G V_c) = \text{Tr}(G) = \sum_i ||A_i||^2.$$

Here we have used the fact that $V_c V_c^T = I$ because V_c is a full rank orthonormal matrix. Thus $J_1^{\text{opt}} = \sum_{i=1}^{n} ||A_i||^2 - \sum_{j=1}^{k} \zeta_j = \sum_{j=k+1}^{c} \zeta_j$.

To see why this is the global optimal solution, we first note that for any solution $\tilde{M}_i, \tilde{R}$, the zero gradient condition holds, i.e, $\tilde{M}_i = A_i^T \tilde{R}$. With this, we have $J_1 = \sum_{i=1}^{n} ||A_i||^2 - \text{Tr}\tilde{R}^T G \tilde{R}$. Due to the positive definiteness of G, the solution for the quadratic function must be unique, up to an arbitrary rotation: $\tilde{R} = R\Gamma$. ■

4 $A_i = L M_i^T$ Decomposition

Theorem 2. The *global* optimal solution for $A_i = L M_i^T$ approximation of J_2 in Eq.(2.9) is given by

$$(4.14) \quad \begin{aligned} L &= U_k, \; M_i = A_i^T U_k, \\ J_1^{\text{opt}} &= \sum_i ||A_i - U_k U_k^T A_i||^2 = \sum_{j=k+1}^{r} \lambda_j. \end{aligned}$$

The proof is identical to Theorem 1, using the relation

$$(4.15) \quad \sum_{j=1}^{r} \lambda_j = \text{Tr}(U_r^T F U_r) = \text{Tr}(F) = \sum_i ||A_i||^2.$$

For this decomposition, when $k = r$, we have $L = I_r$ and $M_i = A_i^T$.

5 $A_i = L M_i R^T$ Decomposition

Theorem 3 The optimal solution for $A_i = L M_i R^T$ approximation of J_3 in Eq.(2.10) is given by

$$(5.16) \quad \begin{aligned} L &= \widetilde{U}_k = (\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k), \\ R &= \widetilde{V}_s = (\tilde{\mathbf{v}}_1, \cdots, \tilde{\mathbf{v}}_s), M_i = \widetilde{U}_k^T A_i \widetilde{V}_s, \end{aligned}$$

where $\tilde{\mathbf{u}}_k, \tilde{\mathbf{v}}_k$ are simultaneous solutions of the eigenvector problems

$$(5.17) \quad \widetilde{F} \tilde{\mathbf{u}}_k = \tilde{\lambda}_k \tilde{\mathbf{u}}_k, \; \widetilde{G} \tilde{\mathbf{v}}_k = \tilde{\zeta}_k \tilde{\mathbf{v}}_k,$$

[2] If eigenvalue $\zeta_j, j \leq k$ is degenerate, the corresponding columns of V_k could be any orthogonal basis of the subspace, therefore not unique.

of the re-weighted covariance matrices $\widetilde{F}$ and $\widetilde{G}$ (see Eq.(1.2)) :

$$\widetilde{F} = \sum_i A_i R R^{\scriptscriptstyle T} A_i^{\scriptscriptstyle T} = \sum_i A_i \widetilde{V}_s \widetilde{V}_s^{\scriptscriptstyle T} A_i^{\scriptscriptstyle T},$$

$$\widetilde{G} = \sum_i A_i^{\scriptscriptstyle T} L L^{\scriptscriptstyle T} A_i = \sum_i A_i^{\scriptscriptstyle T} \widetilde{U}_k \widetilde{U}_k^{\scriptscriptstyle T} A_i. \tag{5.18}$$

The optimal objective function value is given by

$$J_3^{\text{opt}}(k,s) \;=\; \sum_i ||A_i - \widetilde{U}_k \widetilde{U}_k^T A_i \widetilde{V}_s \widetilde{V}_s^T||^2$$

$$= \sum_i ||A_i||^2 - \sum_{j=1}^{k} \tilde{\lambda}_j \tag{5.19}$$

$$\geq \sum_{j=k+1}^{r} \tilde{\lambda}_j + \sum_{j=s+1}^{c} \zeta_j, \tag{5.20}$$

(5.21)

$$J_3^{\text{opt}}(k,s) = \sum_i ||A_i||^2 - \sum_{j=1}^{s} \tilde{\zeta}_j \geq \sum_{j=k+1}^{r} \lambda_j + \sum_{j=s+1}^{c} \tilde{\zeta}_j.$$

In the following special cases, the problem of maximization of J_3 is greatly simplified:
(A) When $k = r$, L becomes a full rank orthogonal matrix. In this case, $LL^T = I_c$, and we can set $L = I_r$. $\tilde{G}$ becomes identical to G. The problem of maximization of J_3 is reduced to the maximization of J_2.
(B) When $s = c$, R becomes a full rank orthogonal matrix. and can be set as $R = I_c$. $\tilde{F}$ becomes identical to F. Maximization of J_3 is reduced to the maximization of J_1.
(C) When $k = r$ and $s = c$, the optimization problem becomes trivial one: $L = I_r, R = I_c, M_i = A_i$.

Proof. We write $J_3 = \sum_{i=1}^n \text{Tr}[A_i^T A_i - 2LM_i R^T A_i^T + M_i^T M_i]$. Taking $\partial J_3/\partial M_i = 0$, we obtain $M_i = L^{\scriptscriptstyle T} A_i R$, and $J_3 = \sum_{i=1}^n ||A_i||^2 - \sum_{i=1}^n ||L^{\scriptscriptstyle T} A_i R||^2$. Thus $\min J_3$ becomes

$$\max_{L,R} J_{3a}(L,R) = \sum_{i=1}^{n} ||L^{\scriptscriptstyle T} A_i R||^2. \tag{5.22}$$

The objective can be written as

$$J_{3a}(L,R) \;=\; \text{Tr}L^{\scriptscriptstyle T}(\sum_{i=1}^{n} A_i R R^{\scriptscriptstyle T} A_i^{\scriptscriptstyle T})L = \text{Tr}L^{\scriptscriptstyle T}\tilde{F}L$$

$$= \text{Tr}R^{\scriptscriptstyle T}(\sum_{i=1}^{n} A_i^T L L^{\scriptscriptstyle T} A_i)R = \text{Tr}R^{\scriptscriptstyle T}\tilde{G}R \tag{5.23}$$

As solutions for these traces of quadratic forms, L, R are given by the eigenvectors of $\widetilde{F}, \widetilde{G}$, and the optimal value are given by the equalities in Eqs.(5.20, 5.21).

To prove the inequality in Eq.(5.20), we note

$$\sum_{j=1}^{r} \tilde{\lambda}_j \;=\; \text{Tr}\widetilde{U}_r^T(\sum_i A_i \widetilde{V}_s \widetilde{V}_s^{\scriptscriptstyle T} A_i^{\scriptscriptstyle T})\widetilde{U}_r \tag{5.24}$$

$$= \text{Tr}\sum_i A_i \widetilde{V}_s \widetilde{V}_s^{\scriptscriptstyle T} A_i^{\scriptscriptstyle T} \tag{5.25}$$

$$= \text{Tr}\widetilde{V}_s^{\scriptscriptstyle T}(\sum_i A_i^{\scriptscriptstyle T} A_i)\widetilde{V}_s \tag{5.26}$$

$$\leq \text{Tr}V_s^{\scriptscriptstyle T}(\sum_i A_i^{\scriptscriptstyle T} A_i)V_s = \sum_{j=1}^{s} \zeta_j \tag{5.27}$$

Re-writing the RHS of above inequality using Eq.(3.13) and splitting the LHS into two terms, we obtain

$$\sum_{j=1}^{k} \tilde{\lambda}_j + \sum_{j=k+1}^{r} \tilde{\lambda}_j \leq \sum_i ||A_i||^2 - \sum_{j=s+1}^{c} \zeta_j.$$

This gives the inequality in Eq.(5.20). The inequality in Eq.(5.21) can be proved in the same fashion. ■

In practice, simultaneous solutions of the $\widetilde{U}, \widetilde{V}$ eigenvectors are achieved via an iterative process:

Iterative Updating Algorithm. Given initial r-by-k matrix $L^{(0)}$, we form $\widetilde{G}$ and solve for the k largest eigenvectors $(\tilde{\mathbf{v}}_1, \cdots, \tilde{\mathbf{v}}_s)$ which gives $R^{(0)}$. Based on $R^{(0)}$, we form $\widetilde{F}$ and solve for the k largest eigenvectors $(\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k)$ which gives $L^{(1)}$. This way, we obtain $L^{(0)}, R^{(0)}, L^{(1)}, R^{(1)}, \cdots$.

Proposition 6. $J_{3a}(L,R)$ is step-wise non-decreasing, i.e.,

$$J_{3a}(L^{(0)}, R^{(0)}) \leq J_{3a}(L^{(1)}, R^{(0)}) \leq J_{3a}(L^{(1)}, R^{(1)}) \leq \cdots.$$

Proof. Suppose we have currently $L^{(t)}, R^{(t)}$. Using $L^{(t)}$ we form $\widetilde{G}$, solve for k largest eigenvectors and obtain a new $R^{(t+1)}$. By definition, $R^{(t+1)}$ is the one that maximizes

$$\text{Tr}R^{\scriptscriptstyle T}(\sum_i A_i^{\scriptscriptstyle T} L^{(t)} {L^{(t)}}^{\scriptscriptstyle T} A_i)R = \sum_i ||L^{(t)T} A_i R||^2.$$

Thus $\sum_i ||L^T A_i R||^2$ must be non-decreasing. Similarly, using $R^{(t)}$ we can form $\tilde{F}$, solve for k largest eigenvectors and obtain a new $L^{(t+1)}$. $\sum_i ||L^T A_i R||^2$ must be also non-decreasing. ■

Proposition 7. An upper-bound exists for $\max \sum_i ||L^T A_i R||^2$:

$$\max_{L \in \mathbb{R}^{r \times k}, R \in \mathbb{R}^{c \times s}} \sum_i ||L^T A_i R||^2 < \min(\sum_{j=1}^{k} \lambda_j, \sum_{j=1}^{s} \zeta_j).$$

Proof. Assume $k < r$. For any r-by-k matrix L, with orthonormal columns, we can always find additional $r - k$

orthonormal columns $\tilde{L}$ such that $(L, \tilde{L})$ span the space. Thus $LL^T + \tilde{L}\tilde{L}^T = I_r$. Noting that $\sum_i A_i^T(\tilde{L}\tilde{L}^T)A_i$ is positive definite, we have

$$\begin{aligned} \max_R \quad & \mathrm{Tr} R^T (\sum_i A_i^T L L^T A_i) R \\ < \quad & \max_R \mathrm{Tr} R^T (\sum_i A_i^T (LL^T + \tilde{L}\tilde{L}^T) A_i) R \\ = \quad & \max_R \mathrm{Tr} R^T (\sum_i A_i^T A_i) R. \end{aligned}$$

From Eqs.(1.2,1.4), the solution to the right-hand-side is given by the 2DSVD: $R = V_s$. We can similarly show the upper-bound involving U_k. The eigenvalues arise from Eqs.(1.2,1.4). ■

From Proposition 6, we obtain a simple lower bound,

$$(5.28) \qquad J_3^{\mathrm{opt}}(k,s) \le \sum_i ||A_i||^2 - \min(\sum_{j=1}^{s} \lambda_j, \sum_{j=1}^{s} \zeta_j)$$

With the non-decreasing property (Proposition 6) and the upper-bound (proposition 7), we conclude that the iterative update algorithm converges to a local maximum.

Is the local maximum also a global maximum? We have several arguments and some strong numerical evidence to support

Observation 8. When $A_i = LM_iR^T$ decomposition provides a good approximation to the 2D data, the iterative update algorithm (IUA) converges to the global maximum.

Discussion. (A) For $n = 1$, 2DSVD reduces to usual SVD and the global maximum is well-known. Fixing L, J_{3a} is a quadratic function of R and the only local maximum is the global one, achieved in IUA. Similarly, fixing R, IUA achieves the global maximum. (B) We may let $L^{(0)} = U_k$ as in 2DSVD, any random matrices, or a matrix of zeroes except one element being 1. For any of these starting point, IUA always converges to the same final solution (L^*, R^*) in 3 iterations. (C) We initialize L as $L^{(0)} \perp L^*$, i.e, as $L^{(0)}$ has zero overlap with the solution L^*. We run IUA again. Typically in 3 iterations, the IUA converges to the same (L^*, R^*).[3] These three experiments indicate it is unlikely IUA can be trapped in a local maximum, if it exists.

5.1 Comparison with $A_i = LM_i, A_i = M_iR^T$

We compare $A_i = LM_iR^T$ with $A_i = M_iR^T$ and $A_i = LM_i^T$. The computer storage for the three approximations are

$$\begin{aligned} &(5.29) \quad S_{\mathrm{LMR}} &=& \; rk + nks + sc = \; 204,000, \\ &(5.30) \quad S_{\mathrm{MR}} &=& \; nrk + kc \qquad = 1,002,000, \\ &(5.31) \quad S_{\mathrm{LM}} &=& \; rk + nkc \qquad = 1,002,000, \end{aligned}$$

where the last number assumes $r = c = 100$, $n = 500$ and $k = s = 20$. The reconstruction errors, i.e., the objective function values, have the relationship:

$$(5.32) \quad J_1^{\mathrm{opt}}(s) < J_3^{\mathrm{opt}}(k,s),\; k < r;\; J_1^{\mathrm{opt}}(s) = J_3^{\mathrm{opt}}(r,s).$$

(5.33)

$$J_2^{\mathrm{opt}}(k) < J_3^{\mathrm{opt}}(k,s),\; s < c;\; J_2^{\mathrm{opt}}(k) = J_3^{\mathrm{opt}}(k,c).$$

This comes from Proposition 7 and noting $J_1^{\mathrm{opt}} = \sum_{j=s+1}^{c} \zeta_j$ and $J_2^{\mathrm{opt}} = \sum_{j=k+1}^{r} \lambda_j$ from Theorems 1 and 2.

From the expressions for J_1^{opt}, J_2^{opt}, and J_3^{opt} in Eqs.(3.12, 4.14), and Theorem 5, we see that A_i is either left projected to the subspace $U_kU_k^T$, right projected to the subspace $V_kV_k^T$ or left and right projected simultaneously.

2DSVD as near-optimal solution for J_3

6 Bounding J_3 by 2DSVD

In this section, we give upper bounds on J_3 and show 2DSVD is the solution for minimizing these upper bounds.

Upper bound J_{3L}

Consider a two-step approximate scheme to solve $\min J_3$.

(L1) We set $A_i \approx LM_iR^T = L(M_iR^T) \equiv LR_i^T$, where $R_i \in \mathbb{R}^{c \times k}$ (not restricted to the special form of M_iR^T), and solve for L, R_i:

$$(6.34) \qquad \min_{L,\, R_i} \sum_{i=1}^{n} ||A_i - LR_i^T||^2.$$

This is identical to $\min J_2$ of Eq.(2.8), and the optimal solution is given by Theorem 2: $L = U_k$, $R_i = A_i^T U_k$.

(L2) We fix L, R_i and find the best approximation of LR_i^T by LM_iR^T, i.e.,

(6.35)

$$\min_{\substack{R,\, M_i \\ L,\, R_i \text{ fixed}}} \sum_i ||LR_i^T - LM_iR^T||^2 = \min_{\substack{R,\, M_i \\ R_i \text{ fixed}}} \sum_i ||R_i - RM_i^T||^2.$$

L drops out since it has orthonormal columns. This is again identical to $\min J_2$ and solution can be obtained. Clearly, the total error is the sum of the two which gives a upper bound for J_3:

$$J_3 \le J_{3L} \equiv \sum_{i=1}^{n} ||A_i - LR_i^T||^2 + \sum_{i=1}^{n} ||LR_i^T - LM_iR^T||^2$$

[3] Due to existence of Γ as discussed in Theorem 1, we measure the angle between the two subspaces. For 1-D subspaces, it is the angle between the two lines. This is generalized to multi-dimensional subspaces [7].

The first term is identical to $\min J_2$, and the optimal solution is given by Theorem 2,

$$(6.36)\quad L \;=\; U_k,\; R_i = A_i^T U_k,\; J_{3L}^{(1)} = \sum_{j=k+1}^{r} \lambda_j.$$

The second term of J_{3L} is equivalent to $\min J_2$, and by Theorem 2 again, optimal solution are given by

(6.37)

$$R = \widehat{V}_s \equiv (\hat{\mathbf{v}}_1, \cdots, \hat{\mathbf{v}}_s),\; M_i = U_k^T A_i R,\; J_{3L}^{(2)} = \sum_{j=k+1}^{c} \hat{\zeta}_j,$$

where $\hat{\mathbf{v}}_k, \hat{\zeta}_k$ are eigenvectors and eigenvalues of the weighted covariance matrix $\widehat{G}$:

$$(6.38)\qquad \widehat{G}\hat{\mathbf{v}}_k = \hat{\zeta}_k \hat{\mathbf{v}}_k,\; \widehat{G} = \sum_i A_i^T U_k U_k^T A_i.$$

Combining these results, we have

Theorem 5. Minimizing the upper bound J_{3L} leads to the following near-optimal solution for J_3:

$$(6.39)\qquad L = U_k,\; R = \widehat{V}_s,\; M_i = U_k^T A_i \widehat{V}_s,$$

$$(6.40)\qquad J_3^{\text{opt}} \le \sum_{j=k+1}^{r} \lambda_j + \sum_{j=s+1}^{c} \hat{\zeta}_j.$$

To implement Theorem 5, we (1) compute U_k; (2) construct the re-weighted row-row covariance $\widehat{G}$ of Eq.(6.38) and compute its s eigenvectors which gives $\widetilde{V}_s$; (3) compute $M_i = U_k^T A_i \widehat{V}_s$. This $U \to V \to M_i$ procedure is a variant of 2DSVD, instead of computing U_k and V_s independent of each other (see Eqs.(1.3, 1.4)). The variant has the same computational cost. We call this LRMi. Note that, in the iterative update algorithm of J_3, if we set $L^{(0)} = U_k$, then $R^{(0)} = \widetilde{V}_k$. This 2DSVD variant can be considered as the initialization of the iterative update algorithm.

Upper bound J_{3R}

Alternatively, we set $A_i \approx L M_i R^T = (L M_i) R^T \equiv L_i R^T$, where $L_i \in \mathbb{R}^{c \times k}$ (not restricted to the special form of $L M_i$). Once R, L_i are computed, we compute the best approximation of $L_i R^T$ by $(L M_i) R^T$. This is equivalent to

$$\min_{L_i,\, R} \sum_{i=1}^{n} ||A_i - L_i R^T||^2 + \min_{\substack{L,\, M_i \\ L_i,\, R \text{ fixed}}} \sum_i ||L_i R^T - L M_i R^T||^2$$

Obviously, this gives a upper bound:

$$J_3 \le J_{3R} \equiv \sum_{i=1}^{n} ||A_i - L_i R^T||^2 + \sum_{i=1}^{n} ||L_i R^T - L M_i R^T||^2$$

R has orthonormal columns and drops out of the second term. The optimization of J_{3R} can be written as

Following the same analysis leading to Theorem 5, we obtain

Theorem 6. Minimizing the upper bound J_{3R} leads to the following near-optimal solution for J_3:

$$(6.41)\qquad L \;=\; \widehat{U}_k \equiv (\hat{\mathbf{u}}_1, \cdots, \hat{\mathbf{u}}_k),\; R = V_s,$$

$$(6.42)\qquad M_i \;=\; \widehat{U}_k^T A_i V_s,$$

$$(6.43)\qquad J_3^{\text{opt}} \;\le\; \sum_{j=k+1}^{r} \hat{\lambda}_j + \sum_{j=s+1}^{c} \zeta_j,$$

where $\tilde{\mathbf{p}}_k$ are eigenvectors of the weighted covariance matrix $\widehat{F}$

$$(6.44)\qquad \widehat{F}\hat{\mathbf{u}}_k = \hat{\zeta}_k \hat{\mathbf{u}}_k,\; \widehat{F} = \sum_i A_i V_s V_s^T A_i^T.$$

The implementations are: (1) compute V_s; (2) construct the re-weighted row-row covariance $\widehat{F}$. of Eq.(6.44) and compute its k eigenvectors which gives $\widetilde{U}_k$; (3) compute M_i. This is another variant of 2DSVD, which we call RLMi.

7 Error Analysis of J_3 and 2DSVD

For $A_i = L M_i R^T$ decomposition, from Theorems 5 and 6, and Eqs.(5.20 , 5.21), we obtain the following lower and upper bounds for J_3:

$$(7.45)\qquad l_b(k,s) \le J_3^{\text{opt}}(k,s) \le u_b(k,s),$$

(7.46)

$$l_b(k,s) = \max(\sum_{j=k+1}^{r} \tilde{\lambda}_j + \sum_{j=s+1}^{c} \zeta_j,\; \sum_{j=k+1}^{r} \lambda_j + \sum_{j=s+1}^{c} \tilde{\zeta}_j),$$

(7.47)

$$u_b(k,s) = \min(\sum_{j=k+1}^{r} \hat{\lambda}_j + \sum_{j=s+1}^{c} \zeta_j,\; \sum_{j=k+1}^{r} \lambda_j + \sum_{j=s+1}^{c} \hat{\zeta}_j).$$

We have seen how 2DSVD arises in minimizing the upper bounds J_{3L} and J_{3R}. Now we analyze it in subspace approximation point of view. Let $\bar{U}_k$ be the subspace complement of $\widetilde{U}_k$, i.e., $(\widetilde{U}_k, \bar{U}_k)$ spans the entire space. Thus $(\widetilde{U}_k, \bar{U}_k)(\widetilde{U}_k, \bar{U}_k)^T = I$. We say that the dominant structures of a 2D map dataset are well captured by the subspace $\widetilde{U}_k \widetilde{U}_k^T$ if

$$\sum_i A_i^T \widetilde{U}_k \widetilde{U}_k^T A_i \simeq \sum_i A_i^T (\widetilde{U}_k \widetilde{U}_k^T + \bar{U}_k \bar{U}_k^T) A_i = \sum_i A_i^T A_i.$$

which will happen if the largest k eigenvalues dominate the spectrum:

$$\sum_{j=1}^{k} \tilde{\lambda}_j \Big/ \sum_{j=1}^{r} \tilde{\lambda}_j \simeq 1, \quad \text{and} \quad \sum_{j=1}^{s} \tilde{\zeta}_j \Big/ \sum_{j=1}^{r} \tilde{\zeta}_j \simeq 1.$$

This is because the importance of these subspaces is approximately measured by their eigenvalues. This situation is similar to the standard SVD, where the first k singular pairs provide a good approximation to the data when

$$\sum_{j=1}^{k} \sigma_j^2 \Big/ \sum_{j=1}^{r} \sigma_j^2 \simeq 1$$

This situation occurs when the eigenvalues λ_j approach zero rapidly with increasing j. The space is dominated by a few eigenstate.

In this case, the 2D maps can be well approximated by the 2DSVD, i.e., 2DSVD provides a near-optimal solution to $J_3(\cdot)$. In this case, the differences between $\hat{\lambda}_j, \tilde{\lambda}_j, \lambda_j$ tend to be small, and we set approximately

$$\sum_{j=k+1}^{r} \hat{\lambda}_j \simeq \sum_{j=k+1}^{r} \tilde{\lambda}_j \simeq \sum_{j=k+1}^{r} \lambda_j .$$

Similar results also hold for $\hat{\zeta}_j, \tilde{\zeta}_j, \zeta_j$. we obtain error estimation,

$$\begin{aligned}
(7.48)\ J_3^{\mathrm{opt}}(k,s) \quad &\simeq \quad \sum_{j=k+1}^{r} \lambda_j + \sum_{j=s+1}^{c} \zeta_j \\
(7.49) \qquad\qquad\qquad &\leq \quad \sum_i ||A_i - U_k U_k^T A_i V_s V_s^T||^2,
\end{aligned}$$

similar to the Eckart-Young Theorem. The two accumulative sums of eigenvalues correspond to the simultaneous left and right projections.

8 $A_i = LM_iL^T$ for symmetric A_i

Consider the case when A_i's are symmetric: $A_i^T = A_i$, for all i. We seek the symmetric decomposition $A_i = LM_iL^T$ of J_4 in Eq.(2.11). Expand J_4 and take $\partial J_4/\partial M_i = 0$, we obtain $M_i = L^{\scriptscriptstyle T} A_i L$, and $J_4 = \sum_{i=1}^n ||A_i||^2 - \sum_{i=1}^n ||L^{\scriptscriptstyle T} A_i^T L||^2$. Thus $\min J_4$ becomes
(8.50)

$$\max_L J_{4a}(L) = \sum_{i=1}^{n} ||L^{\scriptscriptstyle T} A_i L||^2 = \mathrm{Tr} L^{\scriptscriptstyle T} (\sum_{i=1}^{n} A_i L L^{\scriptscriptstyle T} A_i) L$$

Similar to the $A_i = LM_iR^T$ decomposition, 2DSVD gives an near-optimal solution

$$L = U_k,\ M_i = U_k^T A_i U_k. \tag{8.51}$$

Starting with this, the exact optimal solution, can be computed according to the iterative update algorithm in §6. We write
(8.52)

$$\max_{L^{(t+1)}} J_{4a}(L^{(t+1)}) = \mathrm{Tr} L^{(t+1)\scriptscriptstyle T} (\sum_{i=1}^{n} A_i L^{(t)} L^{(t)\scriptscriptstyle T} A_i) L^{(t+1)}.$$

From a current $L^{(t)}$, we form $\widetilde{F} = \sum_i A_i L^{(t)} L^{(t)\scriptscriptstyle T} A_i$ and compute the first k-eigenvectors, which gives $L^{(t+1)}$. From the same analysis of Propositions 6 and 7, we have

$$\begin{aligned}
J_{4a}(L^{(0)}) \quad &\leq \quad J_{4a}(L^{(1)}) \leq J_{4a}(L^{(2)}) \leq \cdots \\
(8.53) \qquad\qquad &\leq \quad \max_L \mathrm{Tr} L^{\scriptscriptstyle T} (\sum_{i=1}^{n} A_i A_i) L = \sum_{i=1}^{n} ||U_k A_i||^2 .
\end{aligned}$$

Thus the iterative algorithm converges to the optimal solution, $L^{(t)} \to \widetilde{U} = (\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k)$, where

$$\widetilde{F} \tilde{\mathbf{u}}_j = \tilde{\lambda}_j \tilde{\mathbf{u}},\ \widetilde{F} = \sum_{i=1}^{n} A_i \widetilde{U}_k \widetilde{U}_k^T A_i . \tag{8.54}$$

The optimal objective value has the lower and upper bounds:

$$\sum_{j=k+1}^{r} (\lambda_j + \tilde{\lambda}_j) \leq J_4^{\mathrm{opt}} \leq \sum_{j=k+1}^{r} (\lambda_j + \hat{\lambda}_j) \tag{8.55}$$

where $\hat{\lambda}_j$ are the eigenvalues of $\widehat{F}$:

$$\widehat{F} \hat{\mathbf{u}}_j = \hat{\lambda}_j \hat{\mathbf{u}},\ \widehat{F} = \sum_{i=1}^{n} A_i U_k U_k^T A_i . \tag{8.56}$$

If eigenvalues $\tilde{\lambda}_j$ fall rapidly as j increases, the principal subspace $\widetilde{U}_k$ captures most of the structure, and 2DSVD provides a good approximation of the data. i.e., 2DSVD is the near-optimal solution in the sense of $J_4(\cdot)$. Thus we have

$$J_4^{\mathrm{opt}} \simeq 2 \sum_{j=k+1}^{r} \lambda_j . \tag{8.57}$$

9 Application to images reconstruction and classification

Dataset A. ORL [4] is a well-known dataset for face recognition. It contains the face images of 40 persons, for a total of 400 images of sizes 92 × 112. The major challenge on this dataset is the variation of the face pose. **Dataset B**. AR [5] is a large face image dataset. The instance of each face may contain large areas of occlusion, due to sun glasses and scarves. The existence of occlusion dramatically increases the within-class variances of AR face image data. We use a subset of AR which contains 65 face images of 5 persons. The original image size is 768 × 576. We crop face part of the image reducing size to 101 × 88.

9.1 Image Reconstruction

Figure 1 shows 8 reconstructed images from the ORL dataset, with a rather small $k = s = 5$. Images in the first row are reconstructed by the $A_i = LM_i$ decomposition using row-row

[4] http://www.uk.research.att.com/facedatabase.html

[5] http://rvl1.ecn.purdue.edu/∼aleix/aleix_face_DB.html

Table 1: Test datasets and related storage for $k = s = 15$.

Dataset	n	Dimensions	# of classes	2DSVD Storage	SVD storage
ORL	400	92×112	40	93060	160560
AR	65	88×101	5	16920	143295

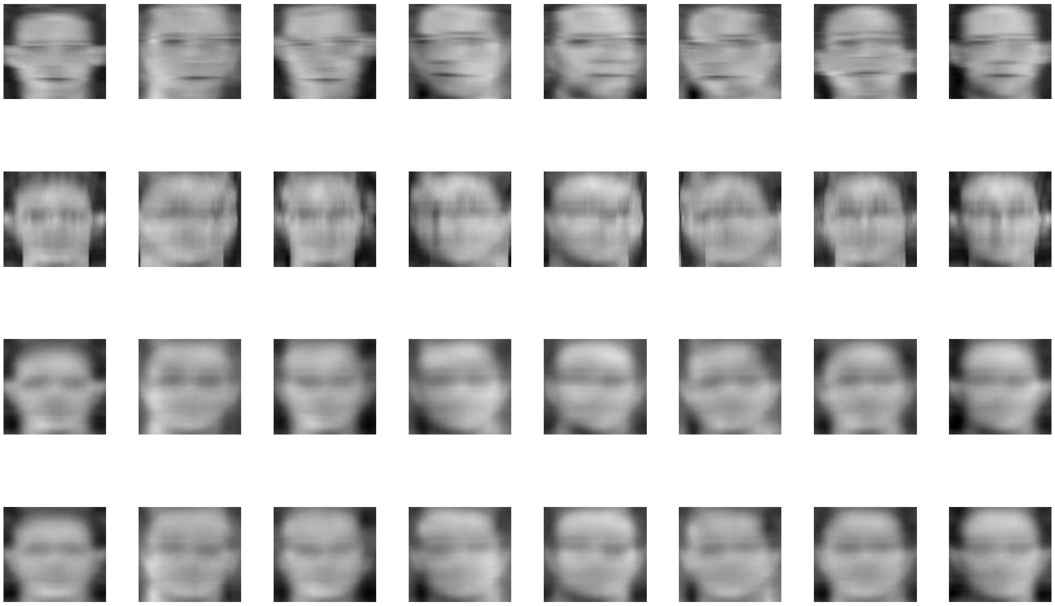


Figure 1: Reconstructed images by 2dLRi (first row), 2dLiR (second row), 2DSVD (third row), and LMR (fourth row) on ORL dataset at $k = s = 5$.

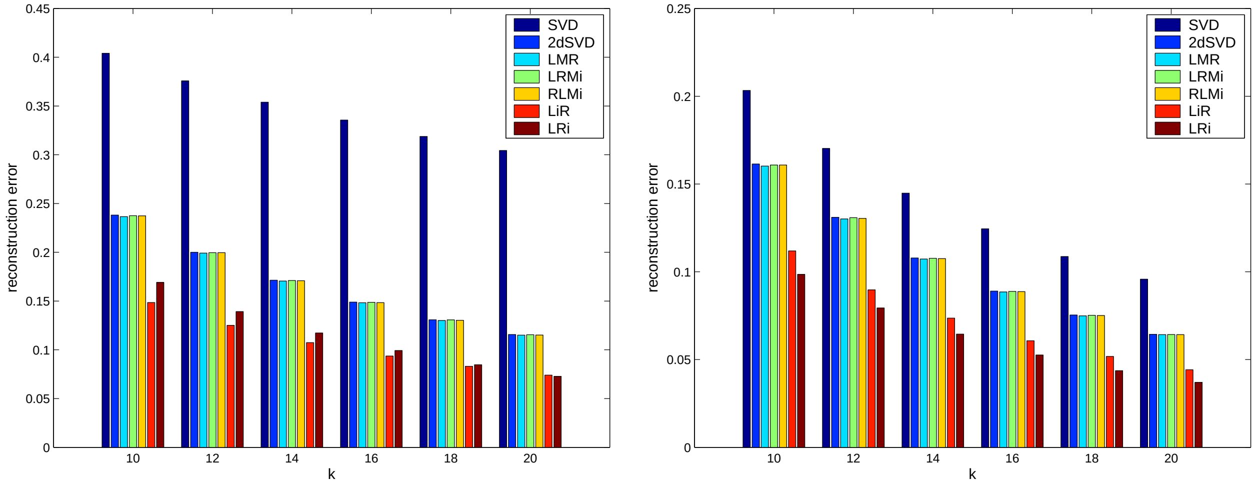


Figure 2: Reconstruction error for ORL dataset (left) and AR dataset (right). Compression methods, from left to right, are indicated in the insert panel.

Table 2: Convergence of LMR

t	2DSVD	Random	Rank-1	Orthogonal
0	0.15889029408304	0.58632462287908	0.99175445002735	0.96285808073065
1	0.15872269104567	0.15872372559621	0.15875108000211	0.15878339838255
2	0.15872268890982	0.15872268893458	0.15872268927498	0.15872268953111
3	0.15872268890976	0.15872268890977	0.15872268890981	0.15872268890981
4	0.15872268890976	0.15872268890976	0.15872268890976	0.15872268890976
angle	0	4.623e-10	3.644e-10	2.797e-10

correlation matrix F. We can see clearly the blurring along horizontal direction. Images in the second row are reconstructed by the $A_i = M_i R$ decomposition using column-column correlation matrix G. We can see clearly the blurring along vertical direction. Images in the 3rd row are reconstructed by the 2DSVD; Images in the 4th row are reconstructed by the LM_iR^T decomposition; The symmetric decomposition of LMR and 2DSVD give better quality reconstruction. Figure 3 shows the same 8 reconstructed images from the ORL dataset, at $k = s = 15$ for 2DSVD and traditional SVD. One can see that 2DSVD gives better quality reconstruction.

Figure 2 shows the reconstruction errors for LMR of §6, 2DSVD, M_iR^T decomposition of §3, LM_i^T decomposition of §4, LRMi of §6, and RLMi of §7. These experiments are done on AR and ORL datasets, with $k = s$ ranging between 10 and 20. We have the following observations: (a) LRi and LiR achieve the lowest residue errors; (b) LMR, 2DSVD, LRMi and RLMi lead to similar residue errors, with LMR the best; (c) SVD has the largest residue errors in all cases.

9.2 Convergence of $A_i = LM_iR^T$ decomposition We examine the sensitivity of LMR on the initial choice. In Table 2, we show J_3 values for several initial choices of $L^{(0)}$ as explained in Discussion of Observation 8: 2DSVD, random matrices, Rank-1 start ($L^{(0)}$ is a matrix of zeros except $L^{(0)}_{1,1} = 1$), and orthogonal start ($L^{(0)}$ is orthogonal to the solution L^*).

We have the following observations. First, starting with all 4 initial $L^{(0)}$'s, the algorithm converges to the same final solution. In the last line, the angle between the different solutions and the one with 2DSVD start are given. They are all around 10^{-10}, practically zero within the accuracy of the computer precision. Considering the rank-1 start and the orthogonal start, this indicates the algorithm does not encounter other local minimums.

Second, 2DSVD is a good approximate solution. It achieves 3 effective decimal digit accuracy: $(J_3(\text{2DSVD}) - J_3^{\text{opt}})/J_3^{\text{opt}} = 0.1\%$. Starting from the 2DSVD, it converges to the final optimal solution in 3 iterations; it gets 6 digits accuracy in 1 iteration and gets 12 digit accuracy in 2 iterations.

Third, the convergence rate is quite good. In 1 iteration, the algorithm converges to 4 digits accuracy for all 4 initial starts. With 4 iterations, the algorithm converges to 14 digits, the computer precision with 64-bits, irrespective of any odd starting points.

To further understand the rapid convergence, we set $k = s = 1$ and run two experiments, one with $L^{(0)} = e_1$ and the other with $L^{(0)} = e_2$, where e_i is a vector of zeroes except that the i-th element is 1. The angle between the solutions at successive iterations, $L_1^{(t)}$ and $L_2^{(t)}$, are given in Table 3. One can see that even though the solution subspaces are orthogonal ($\pi/2$) at beginning, they run towards each other rapidly and become identical in 4 iterations. This indicates the solution subspace converges rapidly.

9.3 Bounds on J_3^{opt} In Figure 4, we show the bounds of J_3^{opt} provided by 2DSVD, Eq.(5.28) and Eq.(7.49). These values are trivially computed once 2DSVD are obtained. Also shown are the exact solutions at $k = s = 10, 15, 20$. We can see the 2DSVD provides a tight upper bound, because it provides a very close optimal solution. This bounds are useful in practice. Suppose one computes 2DSVD and wishes to decide the parameter k and s. Given a tolerance on reconstruction error, one can easily choose the parameters from these bound curves.

9.4 Classification One of the most commonly performed tasks in image processing is the image retrieval. Here we test the classification problem: given a query image, determine its class. We use the K-Nearest-Neighbors (KNN) method based on the Euclidean distance for classification [4, 6]. We have tested $k = 1, 2, 3$ in KNN. $k = 1$ always leads to the best classification results. Thus we fix $k = 1$. We use *10-fold cross-validation* for estimating the classification accuracy. In 10-fold cross-validation, the data are randomly divided into ten subsets of (approximately) equal size. We do the training and testing ten times, each time leaving out one of the subsets for training, and using only the omitted subset for testing. The classification accuracy reported is the average from the ten different random

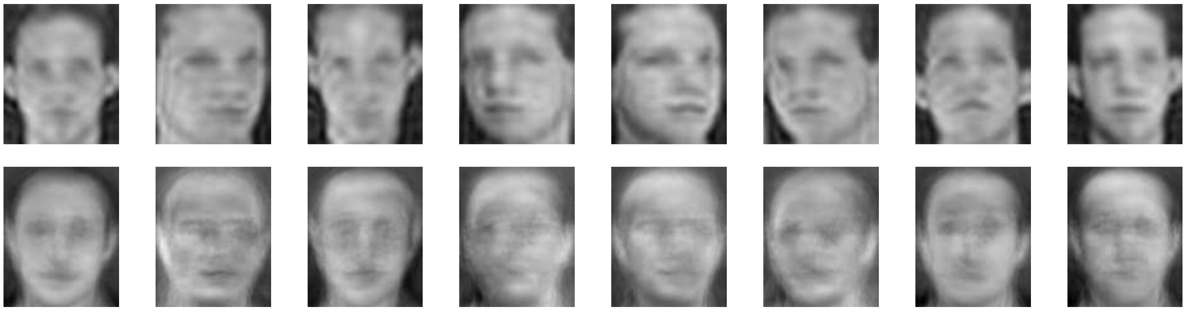


Figure 3: Reconstructed images by 2DSVD (first row), and SVD (second row) on ORL dataset at $k = s = 15$.

Table 3: Convergence of LMR: $k = s = 1$ case

t	0	1	2	3	4
angle	1.571=$\pi/2$	1.486e-03	4.406e-05	1.325e-06	3.985e-08

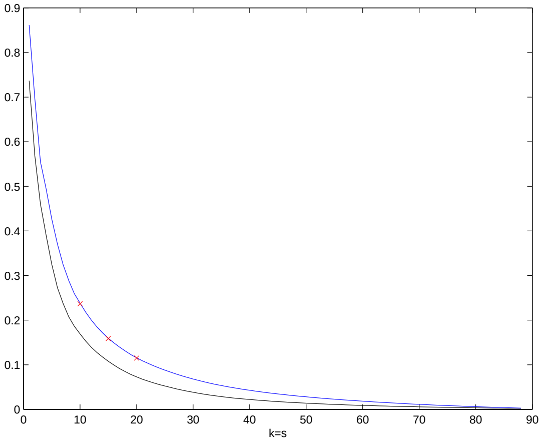


Figure 4: Lower and upper bounds of J_3^{opt} provided by 2DSVD. Also shown are the exact solutions at $k = s = 10, 15, 20$.

splits. The distance between two images are computed using the compressed data:

$$||A_i - A_j|| \approx ||LM_iR^T - LM_jR^T|| = ||M_i - M_j||$$

for LMR, MR, and LM. For SVD, let $(a_i, \cdots, a_n) = U\Sigma(\mathbf{v}_1, \cdots, \mathbf{v}_n)$. The pairwise distance is $||\Sigma(\mathbf{v}_i - \mathbf{v}_j)||$. The results are shown in Fig.5. We see that LMR and 2DSVD consistently leads to small classification error rates, outperforming LiR, LRI and SVD, expect for AR dataset at large value of k (such as $k \geq 16$) where SVD is competitive.

9.5 Convergence for symmetric 2D dataset We tested the algorithm for the symmetric 2D dataset by generating the synthetic datasets $B_i = A_i^T A_i, i = 1, \cdots, n$ for the ORL image dataset. Setting $k = 15$, the reconstruction error J_4 is shown in Table 4. The iteration starts with 2DSVD solution, which is already accurate to 5 digits. After 1 iteration, the algorithm converges to the machine precision.

Table 4: Convergence for symmetric case

t	J_4
0	0.01245341543106
1	0.01245337811927
2	0.01245337811927

10 Surface temperature maps

The datasets are 12 maps, each of size 32 (latitude) x 64 (longitude). Each shows the distribution of average surface temperature of the month of January (100 years).

Table 5 shows the reconstruction of the temperature maps. One see that 2DSVD provides about the same or better reconstruction at much less storage. This shows 2DSVD provides a more effective function approximation of these 2D maps. The temperature maps are shown in Figure 6.

11 Summary

In this paper, we propose an extension of standard SVD for a set of vectors to 2DSVD for a set of 2D objects $\{A_i\}_{i=1}^n$. The resulting 2DSVD has a number of optimality properties which make it suitable for low-rank approximation. We systematically analyze the four decompositions, $A_i = M_iR^T$, $A_i = LM_i^T$, $A_i =$

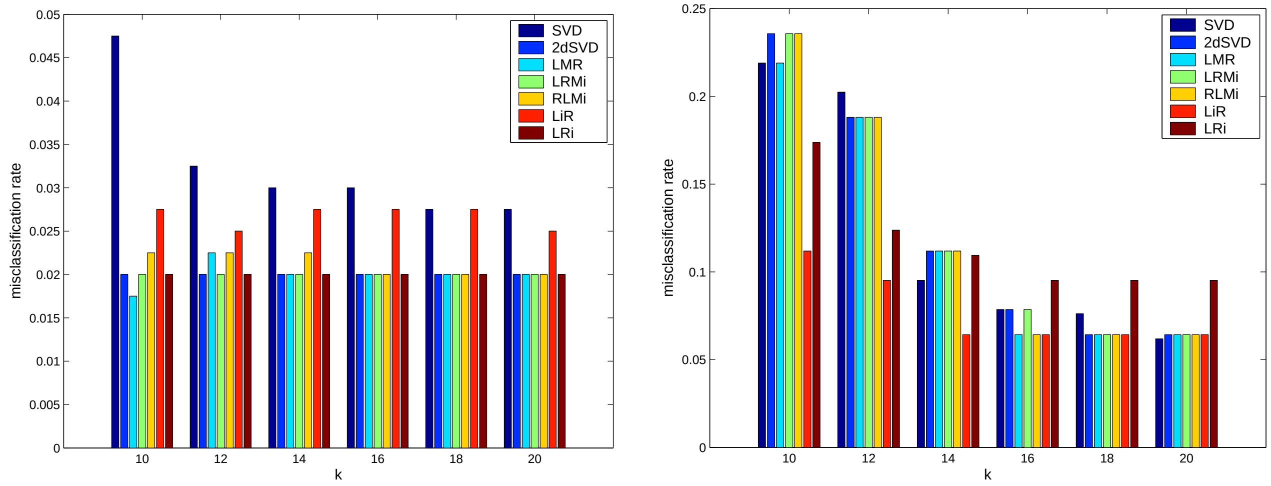


Figure 5: Classification (cross validation) error rate for ORL (left) and for AR (right)

Table 5: Reconstruction of the temperature maps

Method	k,s	storage	error
2DSVD	$k = 4, s = 8$	1024	0.0030
2DSVD	$k = 8, s = 16$	2816	0.0022
SVD	$k = 4$	8244	0.0040
SVD	$k = 8$	16488	0.0022

LM_iR^T, and $A_i = LM_iL^T$ for symmetric A_i. Their relationship with 2DSVD are shown. This provides a framework unifying two recent approaches by Yang *et al.*[13] and by Ye [14] for low-rank approximations which captures explicitly the 2D nature of the 2D objects, and further extend the analysis results. We carry out extensive experiment on 2 image datasets and compare to standard SVD. We also apply 2DSVD to weather maps. These experiments demonstrate the usefulness of 2DSVD.

Acknowledgments. We thank Dr. Haesun Park for discussions. This work is partially supported by Department of Energy under contract DE-AC03-76SF00098.

References

[1] M.W. Berry, S.T. Dumais, and Gavin W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37:573–595, 1995.

[2] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the Society for Information Scienc*, *41*, 391–407.

[3] Dhillon, I., & Modha, D. (2001). Concept decompositions for large sparse text data using clustering. *Machine Learning*, *42*, 143–175.

[4] R.O. Duda, P.E. Hart, and D. Stork. *Pattern Classification*. Wiley, 2000.

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

[6] K. Fukunaga. *Introduction to Statistical Pattern Classification*. Academic Press, San Diego, California, USA, 1990.

[7] G. Golub and C. Van Loan. *Matrix Computations, 3rd edition*. Johns Hopkins, Baltimore, 1996.

[8] I.T. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2002.

[9] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Trans. Pattern Analysis Machine Intelligence*, 12:103–108, 1990.

[10] T.G. Kolda. Orthogonal tensor decompositions. *SIAM J. Matrix Analysis and App.*, 23:243–255, 2001.

[11] R. W. Preisendorfer and C. D. Mobley. *Principal Component Analysis in Meteorology and Oceanography*. Elsevier Science Ltd, 1988.

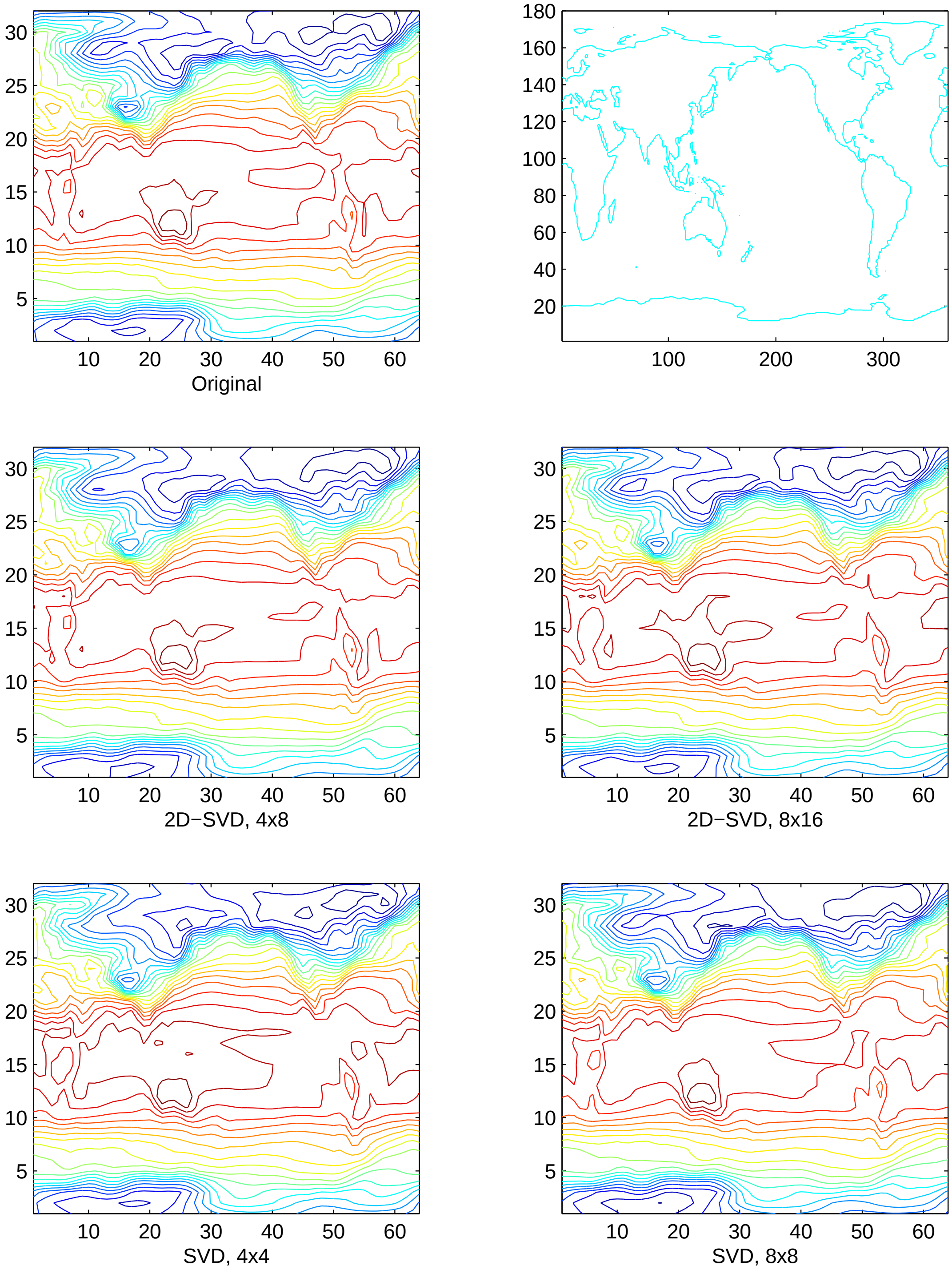
[12] N. Srebro & T. Jaakkola. Weighted low-rank approximations. *ICML Conference Proceedings* (pp. 720–727).

[13] J. Yang, D. Zhang, A. Frangi, and J. Yang. Two-dimensional pca: A new approach to appearance-based face representation and recognition. *IEEE Trans. Pattern Analysis Machine Intelligence*, 26:131–137, 2004.

[14] J. Ye. Generalized Low Rank Approximations of Matrices. *Proceedings of the Twenty-First International Conference on Machine Learning*. 887–894, 2004.

[15] J. Ye, R. Janardan, and Q. Li. GPCA: An Efficient Dimension Reduction Scheme for Image Compression and Retrieval. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 354–363, 2004.

[16] T. Zhang and G. H. Golub. Rank-one approximation to high order tensors. *SIAM Journal of Matrix Analysis and Applications*, 23:534–550, 2001.



Original

2D−SVD, 4x8

2D−SVD, 8x16

SVD, 4x4

SVD, 8x8

Figure 6: Global surface temperature. Top left: original data (January temperature for a randomly picked year. One can see that the central area of Australia is hottest spot on Earth). Top right: matching continental topography for location specification. Middle left: 2DSVD with $k = 4, s = 8$. The reduction ratio are kept same for both columns and rows: 8=32/4=64/8. Middle right: 2D-SVd with $k = 8, s = 16$. Bottom left: conventional SVD with $k = 4$. Bottom right: conventional SVD with $k = 8$.