Constrained Quadratic Optimization Problems with Applications

Mohammed A. Hasan
Department of Electrical & Computer Engineering
University of Minnesota Duluth
E.mail:mhasan@d.umn.edu

Abstract

Optimization problems involving strictly quadratic constraints arise in many algorithmic developments in signal processing, applied mathematics, physics, and control theory. In this paper, a framework is developed based on 1) incorporating diagonal matrices into the cost function and/or the constraints, and 2) exploiting the symmetry of quadratic constraints in the Lagrangian function. Using this framework, many algorithms for true principal and minor component extraction and true principal singular component analysis are derived from optimizing a weighted inverse Rayleigh quotient and weighted Rayleigh quotient like criteria. The main features of these algorithms are that they are self-stabilizing, can compute multiple principal, minor or singular components, and they ensure orthogonality. Additionally, using a logarithmic cost function, fast convergent power-like methods for computing principal components and singular vectors are developed. Three lists for minor component analysis, principal component analysis, and principal singular component analysis are provided.

Keywords: subspace methods, power method, bi-iteration, information criteria, self-normalizing neural networks, Eigenvalue problem, learning algorithms, minor component analysis (MCA), principal component analysis (PCA), principal singular component analysis (PSCA), singular value decomposition (SVD), principal singular subspace (PSS), gradient algorithms.

1. Introduction

Finding global minima and maxima of constrained optimization problems is an important task in engineering applications and scientific computation. For example, eigendecomposition and singular value decomposition, which are basic tools in many algorithms can be obtained by optimization some criteria over quadratic constraints. In this paper, the main focus will be principal, minor component estimation, and principal singular component analysis.

The goal in minor component analysis (MCA) is to determine the directions of smallest variance in a distribution. These directions correspond to the eigenvectors of the covariance matrix of the data which have the smallest eigenvalues. The principal component analysis (PCA) deals with the recovery of the eigenvectors associated with the largest eigenvalues of the autocorrelation matrix of the input data. A comprehensive study of single minor component analysis is given in [1]. Algorithms for multiple MCA and PCA extraction have been developed in [2]-[6]. The goal of singular subspace analysis is to track or estimate the principal or minor singular (left or right) subspaces of a sequence of random vectors. Several subspace extraction algorithms for this problem have been proposed in the literature. Gradient flows for learning SVD are proposed in [7]-[9]. A logarithmic cost function is used in [10] for tracking singular subspaces.

Neural network techniques for principal component analysis (PCA) have been extensively researched, while learning rules for minor component analysis and singular component analysis are not fully developed. Thus the main focus is to develop MCA and SVD learning rules derived directly from certain cost functions.

The main drawback of some MCA [5], PCA [6], and PSS algorithms [10] is that these algorithms can only produce an arbitrary orthonormal basis of the principal singular subspace. In this paper, the proposed PCA, MCA and PSS algorithms determine exactly the desired eigenvectors or singular vectors and their corresponding eigenvalues and singular values, respectively. The key idea of this work is based on exploiting the symmetry of the constraints when forming the Lagrangian of the problem. This approach turns out to be effective in deriving learning rules that are similar to learning rules based on the concept of natural gradient.

2. General Problem Formulation

Consider the following minimization problem

$$\text{Optimize} \quad F(x) \ \text{ subject to } x^T B x = D, \tag{1}$$

where F is at least twice continuously differentiable real valued function, $x \in \mathbb{R}^{n\times r}$, $B \in \mathbb{R}^{n\times n}$ is positive semidefinite, and D a positive definite matrix is a diagonal matrix of size r. Here $tr(X)$ denotes the trace of a square matrix X, $(.)^T$ denotes matrix transpose. Define the Lagrangian of (1) as

$$\begin{aligned} \mathcal{L}(x,\lambda) &= F(x) - tr\{(x^T B x - D)\frac{\lambda}{2}\}, \\ &= F(x) - tr\{(x^T B x - D)\frac{\lambda^T}{2}\}, \end{aligned} \tag{2}$$

where λ is a matrix of Lagrange multipliers. The second equation holds since the constraint function $x^T Bx - D$ is symmetric.

The first order necessary condition for optimality is that $\nabla\mathcal{L} = 0$, where

$$\nabla\mathcal{L} = \begin{Bmatrix} \nabla_x F(x) - Bx\lambda \\ x^T Bx - D \end{Bmatrix} = \begin{Bmatrix} \nabla_x F(x) - Bx\lambda^T \\ x^T Bx - D \end{Bmatrix}. \tag{3a}$$

Thus for any critical point x for which $\nabla\mathcal{L}(x,\lambda) = 0$, the Lagrange multiplier matrix λ is symmetric, i.e., $\lambda = \lambda^T$. If x is an optimal solution for (1), then λ may be expressed as

$$\lambda = D^{-1}x^T\nabla_x F(x) = \nabla_x F(x)^T x D^{-1}, \tag{3b}$$

Substituting these expressions for λ in (3a) yields

$$\begin{aligned} \nabla_x\mathcal{L} &= \nabla_x F(x) - Bxx^T\nabla_x F(x) \\ &= (I - BxD^{-1}x^T)\nabla_x F(x), \\ &= \nabla_x F(x) - Bx\nabla_x F(x)^T x D^{-1}. \end{aligned} \tag{4}$$

If $B = I$ and $D = I$, where I is an identity matrix of appropriate dimension, then the expression $\nabla_x\mathcal{L} = \nabla_{NG}F$, where $\nabla_{NG}F$ represents the natural gradient of F over the orthogonality constraint $x^T x = I$ [11]-[12]. Note that if $\nabla_x\mathcal{L} = 0$, and λ is nonsingular, then $x^T\nabla_x\mathcal{L} = (I - x^T BxD^{-1})x^T\nabla F(x)$. This implies that $x^T Bx = D$. Consequently, if $\nabla_x\mathcal{L} = (I - BxD^{-1}x^T)\nabla_x F(x)$ is forced to be zero by a gradient method, then the resulting solution x satisfies the constraints $x^T Bx = D$. The matrix $P = I - BxD^{-1}x^T$ defines a projection onto the region defined by the constraint $x^T Bx = D$ in that $x^T P = x^T$ for any x satisfying $x^T Bx = D$.

3. Principal & Minor Component Analysis

In this section we develop gradient flows that are capable of extracting the principal and minor subspace and the minor eigenvectors from the optimization of a weighted Rayleigh quotient (WRQ) and a weighted inverse Rayleigh quotient (WIRQ). WRQ and WIRQ has several attractive properties that can be exploited for deriving MCA and PCA algorithms. These include boundedness, homogeneity and some orthogonality properties. Additionally, as will be shown later, under mild conditions each of WRQ and WIRQ has only one minima and one maxima.

Suppose that the input vector sequence $x_k \in \mathbb{R}^n$ is a stationary stochastic process with zero mean and covariance matrix $B = E(xx^T)$ with the eigenvalues $0 < \lambda_1 < \cdots < \lambda_n$ and the corresponding orthonormal eigenvectors $z_1, \cdots, z_n$. Let p be an integer such that $1 \leq r \leq n$ and let the eigendecomposition of B be given as $B = Z_1\Lambda_1 Z_1^T + Z_2\Lambda_2 Z_2^T$, where $Z_1 = [z_1, \cdots, z_r]$, $\Lambda_1 = diag\{\lambda_1, \cdots, \lambda_r\}$, $\Lambda_2 = diag\{\lambda_{r+1}, \cdots, \lambda_n\}$, and $Z_2 = [z_{r+1}, \cdots, z_n]$. The main objective is to compute the true MCA and PCA of dimension r, i.e., true MCA is to find the r $(1 \leq r \leq n)$ smallest eigenvalues $\lambda_1, \cdots, \lambda_r$, and corresponding eigenvectors $z_1, \cdots, z_r$. Similarly, true PCA is to find the r largest eigenvalues $\lambda_{n-r+1}, \cdots, \lambda_n$, and corresponding eigenvectors $z_{n-r+1}, \cdots, z_n$.

The PCA and MCA development will start with the basic unconstrained optimization of weighted Rayleigh quotient WRQ and weighted inverse Rayleigh quotient WIRQ. The WRQ is given by

$$G_1(U) = tr\{(U^TU)^{\frac{-1}{2}}U^TBU(U^TU)^{\frac{-1}{2}}D\}, \tag{5}$$

while the WIRQ is defined as

$$G_2(U) = tr\{(U^TBU)^{\frac{-1}{2}}U^TU(U^TBU)^{\frac{-1}{2}}D\}. \tag{6}$$

A closed form for the gradient of these criteria may not be easy to determine. Thus we will consider the following criteria for WRQ and WIRQ, respectively:

$$\text{Optimize}\{F_1(U) = tr\{(U^TBU)(U^TU)^{-1}D\}, \tag{7}$$

and

$$\text{Optimize}\{F_2(U) = tr\{(U^TU)(U^TBU)^{-1}D\}, \tag{8}$$

over all full rank matrices $U \in \mathbb{R}^{n\times r}$. Here D is a diagonal matrix of size r having distinct positive eigenvalues. It will be assumed that $D = \text{diag}(d_1, \cdots, d_r)$ and that $d_1 > d_2 > \cdots > d_r > 0$.

The gradient of F_1 and F_2 are

$$\begin{aligned}\nabla F_1 &= BU(U^TU)^{-1}D + BUD(U^TU)^{-1} - BU(U^TU)^{-1}\\ &\times DU^TBU(U^TU)^{-1} - BU(U^TU)^{-1}U^TBUD(U^TU)^{-1},\\ \nabla F_2 &= U(U^TBU)^{-1}D + UD(U^TBU)^{-1} - BU(U^TBU)^{-1}\\ &\times DU^TU(U^TBU)^{-1} - BU(U^TBU)^{-1}U^TUD(U^TBU)^{-1}.\end{aligned} \tag{9, 10}$$

The following proposition deals with the critical points of WRQ and WIRQ.

Proposition 1 (Stationarity). *Let D be a diagonal matrix such that the diagonal entries of D are positive, distinct, and arranged in descending order and let B be a real symmetric n-dimensional matrix with eigenvalues $0 < \lambda_1 < \cdots < \lambda_r < \lambda_{r+1} < \cdots < \lambda_n$ and the corresponding orthonormal eigenvectors $z_1, \cdots, z_n$. Then*

$$\max\{F_1(U)\} = \sum_{k=1}^{r} d_k\lambda_{n-k+1},\ \min\{F_1(U)\} = \sum_{k=1}^{r} d_k\lambda_k, \tag{11a}$$

$$\max\{F_2(U)\} = \sum_{k=1}^{r} \frac{d_k}{\lambda_k},\ \min\{F_2(U)\} = \sum_{k=1}^{r} \frac{d_k}{\lambda_{n-k+1}}. \tag{11b}$$

Moreover, the global minimum and the global maximum of F_1 are attained if and only if $U = Z_1\Pi_1$ and $U = Z_2\Pi_2$, respectively, where $Z_1 = [z_{n-r+1}\cdots z_n]$ and $Z_2 = [z_1\cdots z_r]$ and Π_1, Π_2 are permutation matrices. Similarly, the global minimum and the global maximum of F_2 are attained if and only if $U = Z_2\Pi_2$ and $U = Z_1\Pi_1$, respectively. All other critical points are saddles.

Outline of Proof: Let $U = ZE$, where Z is any matrix consisting of r eigenvectors, and E is a nonsingular matrix, then

$$\begin{aligned}F(ZE) &= tr((E^TE)(E^T\Lambda E)^{-1}D)\\ &= tr(E^T\Lambda^{-1}E^{-T}D),\end{aligned} \tag{12}$$

where $\Lambda = diag(\lambda_{i_1}\cdots\lambda_{i_r})$ and $(i_1, \cdots \lambda_{i_r})$ is a permutation of $\{1, \cdots, n\}$. The possible maximum or minimum of $F(U)$ occurs when $\nabla_E tr(WIRQ(ZE, B, D)) = \nabla_E F(ZE) = 0$. It can be shown that

$$\nabla_E F(ZE) = -E^{-T}DE^T\Lambda^{-1}E^{-T} + \Lambda^{-1}E^{-T}D. \tag{13}$$

This implies that $E^T\Lambda^{-1}E^{-T}D = DE^T\Lambda^{-1}E^{-T}$. Since D is diagonal with distinct eigenvalues, it follows from Proposition 4 (see Appendix) that $E^T\Lambda_r^{-1}E^{-T}$ is diagonal. Thus the only possible solution of $\nabla_E tr(F(ZE)) = 0$ is that $E = D_1P$, where D_1 is diagonal, and P is a permutation matrix. Now, at stationarity points the objective function is given by

$$F(E) = F(D_1P) = tr(P\Lambda^{-1}P^TD). \tag{14}$$

Clearly, since the diagonal entries of D are in descending order, then among all possible Λ and all possible permutations P, the maximum of $F_1(U)$ occurs at $\Lambda = \text{diag}(\lambda_{n-r+1}, \cdots, \lambda_n)$ and $P = I$. Similarly, the minimum occurs at $\Lambda = \Lambda_1$ and $P = J$, where J is the interchange matrix given by $J = [e_r, e_{r-1}\cdots e_1]$ where e_i is the ith column of an $r \times r$ identity matrix. To examine the critical points for maxima and minima, we have to show that the Hessian matrix defined as $H\phi(U) = \frac{\partial}{(vecU)^T}\left(\frac{\partial\phi(U)}{\partial(vecU)^T}\right)^T$, where $\phi(U) = tr(F(U))$, is positive semi-definite at $U = Z_1$ and negative semi-definite at $U = Z_2$. Here vec stands for the operation of stacking the columns of a matrix into one column. It is non definite at any other critical points.

Q. E. D.

Proposition 1 indicates that with a properly chosen D, $F_1(U)$ and $F_2(U)$ have exactly one global minima and one global maxima.

3.1 Gradient Flows

The ordinary differential equation (ODE) associated with the gradient systems (9) and (10) are:

$$\begin{aligned}U' = \nabla F &= U\{(U^TBU)^{-1}D + D(U^TBU)^{-1}\}\\ &- BU(U^TBU)^{-1}\{DU^TU + U^TUD\}(U^TBU)^{-1},\end{aligned} \tag{16}$$

$$\begin{aligned}U' = \nabla F &= BU\{(U^TU)^{-1}D + D(U^TU)^{-1}\}\\ &- BU(U^TU)^{-1}\{DU^TBU + U^TBUD\}(U^TU)^{-1},\end{aligned} \tag{17}$$

where $U'(t) = \frac{dU(t)}{dt}$.

To alleviate matrix inversion, the quadratic constraint $U^TBU = I$ is imposed in (16) so that for any U satisfying $U^TBU = I$ we have

$$U' = \nabla F = 2UD - BU\{DU^TU + U^TUD\}. \tag{18a}$$

Similarly, the quadratic constraint $U^TU = I$ is imposed in (17) to obtain:

$$U' = \nabla F_2 = 2BUD - U\{DU^TBU + U^TBUD\}. \tag{18b}$$

In the next proposition, we show that under a mild condition, the gradient ascent with sufficiently small step-size converges to the true MCA.

Proposition 2. *Let D and B be as in Proposition 1 and let U_∞ be the solution of the difference equation*

$$U_{k+1} = U_k + \alpha\{U_kD - \frac{1}{2}BU_k(DU_k^TU_k + U_k^TU_kD\}, \tag{19}$$

for some learning step size $\alpha \in (0, 1)$. Assume that $DU_\infty^TU_\infty + U_\infty^TU_\infty D$ is non-singular. Then the limiting solution U_∞ of the gradient ascent iteration (19) satisfies the following:

1. *$U_\infty^TBU_\infty = I$, and $U_\infty^TU_\infty = \Lambda_1^{-1}$*
2. *$U_\infty = Z_1\Lambda_1^{\frac{-1}{2}}$ and $F(U_\infty) = \sum_{k=1}^{p}\frac{d_k}{\lambda_k}$*

Proof: Since there is only one maxima, then for any initial matrix U_0 satisfying $U_0^TBU_0 = I$ the gradient ascent (19) converges globally to system's equilibrium point. Assume that U_∞ is the limiting solution of the gradient ascent iteration (19),

then $U_\infty^T U_\infty D = U_\infty^T B U_\infty H$, where $H = U_\infty^T U_\infty D + D U_\infty^T U_\infty$. Clearly,

$$2H = H U_\infty^T B U_\infty + U_\infty^T B U_\infty H. \quad (20)$$

We show next that each eigenvalue of $U_\infty^T B U_\infty$ is equal to 1. Let λ be an eigenvalue of $U_\infty^T B U_\infty$ with corresponding eigenvector x, then $U^T B U x = \lambda x$. By post-multiplying and pre-multiplying both sides of (20) by x and x^T, respectively we obtain $2x^T H x = \lambda x^T H x + \lambda x^T H x$ and thus $(1-\lambda) x^T H x = 0$. The nonsingularity of H implies that $\lambda = 1$. Since $U_\infty^T B U_\infty$ is symmetric then $B = I$. Consequently, $U_\infty^T U_\infty D = D U_\infty^T U_\infty$. Since D is diagonal with distinct eigenvalues, we have from Proposition 4 (see Appendix) it follows that $U_\infty^T U_\infty$ is diagonal. This shows that $U_\infty^T U_\infty = \Lambda_1^{-1}$ and $B U_\infty^T = U_\infty (U_\infty^T U_\infty)^{-1} = U_\infty \Lambda_1^{-1}$. Consequently, $U_\infty = Z_1 \Lambda_1^{\frac{-1}{2}}$.

Q. E. D.

The gradient flow (18b) can be analyzed analogously.

3.2 Constrained Optimization

The unconstrained optimization problem (7) can be converted to the following constrained optimization problems:

$$\text{Optimize } tr(U^T U D) \text{ subject to } U^T B U = I, \quad (21a)$$

$$\text{Optimize } tr(U^T U) \text{ subject to } U^T B U = D, \quad (21b)$$

$$\text{Optimize } tr(U^T U D_1) \text{ subject to } U^T B U = D_2,\ D_1 D_2 = D, \quad (21c)$$

$$\text{Optimize } tr(U^T B U)^{-1} \text{ subject to } U^T U = D^{-1}, \quad (21d)$$

$$\text{Optimize } tr((U^T B U)^{-1} D) \text{ subject to } U^T U = I, \quad (21e)$$

$$\text{Optimize } tr((U^T B U)^{-1} D_1) \text{ subject to } U^T U = D_2,\ D_1 D_2 = D. \quad (21f)$$

The constrained optimization method developed in Section 2 will be applied to the above problems. The resulting MCA and PCA flows are given in the following algorithms:

3.2.1 MCA Algorithms

$$U_{k+1} = U_k + \alpha(U_k D - B U_k D U_k^T U_k) \quad (22a)$$
$$U_{k+1} = U_k + \alpha(U_k - B U_k U_k^T U_k D^{-1}) \quad (22b)$$
$$U_{k+1} = U_k + \alpha(U_k D_1 - B U_k D_1 U_k^T U_k D_2^{-1}) \quad (22c)$$
$$U_{k+1} = U_k + \alpha(-B U_k (U_k^T B U_k)^{-2} + U_k (U_k^T B U_k)^{-1} D) \quad (22d)$$
$$U_{k+1} = U_k + \alpha(-B U_k (U_k^T B U_k)^{-1} D (U_k^T B U_k)^{-1} + U_k (U_k^T B U_k)^{-1} D) \quad (22e)$$
$$U_{k+1} = U_k + \alpha(-B U_k (U_k^T B U_k)^{-1} D_1 (U_k^T B U_k)^{-1} + U_k (U_k^T B U_k)^{-1} D_1 D_2^{-1}) \quad (22f)$$

Similarly, the unconstrained optimization problem (8) can be converted to the following constrained problems:

$$\text{Optimize } tr(U^T B U D) \text{ subject to } U^T U = I, \quad (23a)$$

$$\text{Optimize } tr(U^T B U) \text{ subject to } U^T U = D, \quad (23b)$$

$$\text{Optimize } tr(U^T B U D_1) \text{ subject to } U^T U = D_2^{-1},\ D_1 D_2 = D \quad (23c)$$

$$\text{Optimize } tr(U^T U)^{-1} \text{ subject to } U^T B U = D^{-1}, \quad (23d)$$

$$\text{Optimize } tr((U^T U)^{-1} D) \text{ subject to } U^T B U = I, \quad (23e)$$

$$\text{Optimize } tr((U^T U)^{-1} D_1) \text{ subject to } U^T B U = D_2, \quad (23f)$$

The constrained optimization techniques of Section 2 yield the following PCA flows:

3.2.2 PCA Algorithms

$$U_{k+1} = U_k + \alpha(B U_k D - U_k D U_k^T B U_k) \quad (24a)$$
$$U_{k+1} = U_k + \alpha(B U_k - U_k U_k^T B U_k D^{-1}) \quad (24b)$$
$$U_{k+1} = U_k + \alpha(B U_k D_1 - U_k D_1 U_k^T B U_k D_2) \quad (24c)$$
$$U_{k+1} = U_k + \alpha(-U_k (U_k^T U_k)^{-2} + B U_k (U_k^T U_k)^{-1} D) \quad (24d)$$
$$U_{k+1} = U_k + \alpha(-U_k (U_k^T U_k)^{-1} D (U_k^T U_k)^{-1} + B U_k (U_k^T U_k)^{-1} D) \quad (24e)$$
$$U_{k+1} = U_k + \alpha(-U_k (U_k^T U_k)^{-1} D_1 (U_k^T U_k)^{-1} + B U_k (U_k^T U_k)^{-1} D_1 D_2^{-1}) \quad (24f)$$

Simulations showed that many of these iterations such as (22a, 22b) and (24a, 24b) respectively converge to diagonal MCA and PCA with or without normalization using a positive learning rate, i.e., they are self-normalizing neural networks. However, if negative learning rate is used, these algorithms converge to PCA and MCA, respectively only if normalization is used.

3.2.3 Logarithmic Cost Functions

True PCA and MCA algorithms can also be derived using logarithmic cost functions. We will consider the following optimization problems:

$$\text{Optimize } tr(\log(U^T B U + D) \text{ subject to } U^T U = I, \quad (25a)$$

$$\text{Optimize } tr(\log(U^T U + D)) \text{ subject to } U^T B U = D. \quad (25b)$$

Since the constraint functions are symmetric, the theory of Section 2 can be applied to obtain the following flows:

$$U' = B U (U^T B U + D)^{-1} + U (U^T B U + D)^{-1} D - U, \quad (25c)$$

$$U' = U (U^T U + D)^{-1} + B U (U^T U + D)^{-1} D - B U. \quad (25d)$$

Numerical simulations showed that the flows of (26a) and (26b) converge to the actual MCA and PCA respectively using a positive learning rate. Analogous learning rules which are slightly different from those in (26) were derived in [6], [13] where unconstrained logarithmic functions of the form $tr(\log(U^T B U) - tr(U^T U)$ [6] or $tr(\log(U^T U) - tr(U^T B U)$ [13] were considered. The main difference is that the learning rules in [6] and [13] find principal or minor subspaces but not the actual eigenvectors. One can also consider the following unconstrained optimization problems:

$$\text{Optimize } tr(\log(U^T B U + D) - tr(U^T U), \quad (26a)$$

$$\text{Optimize } tr(\log(U^T U + D)) - tr(U^T B U). \quad (26b)$$

The gradient flows corresponding to the cost functions (26a) and (26b) are

$$U' = B U (U^T B U + D)^{-1} - U, \text{ PCA Flow} \quad (26c)$$

$$U' = U (U^T U + D)^{-1} - B U, \text{ MCA Flow.} \quad (26d)$$

3.2.4 Power-Like PCA Algorithm

The learning rules (26) can be slightly modified to obtain the following power like iterations:

$$U_{k+1} = B U_k (U_k^T B U_k + D)^{-1}, \quad (27a)$$

$$U_{k+1} = B U_k (U_k^T U_k + D)^{-1}. \quad (27b)$$

It can be shown that in the limit both $U_k^T U_k$ and $U_k^T B U_k$ converge to diagonal matrices as $k \to \infty$. Let $P = U_\infty^T U_\infty$ and $Q = U_\infty^T B U_\infty$, then $P = Q(Q+D)^{-1}$ or $P(Q+D) = Q$. Since Q and P are symmetric, if follows that $QP = PQ$ and $P(Q+D) = (Q+D)P$. This implies that $PD = DP$ and hence P (Proposition 4 see Appendix) is diagonal matrix. Therefore, Q is diagonal provided that all eigenvalues of P are distinct.

4. Principal Singular Component Analysis

Let $A \in \mathbb{R}^{m \times n}$ be a rectangular matrix. The SVD of the matrix A is written as

$$A = F \Sigma G^T = \sum_{k=1}^{p} \sigma_k f_k g_k^T,$$

where $p = \min\{m, n\}$, and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p$. The singular vectors corresponding to the null space of A are not included in this decomposition. Here $F = [f_1, f_2, \cdots, f_p]$ and $V = [g_1, g_2, \cdots, g_p]$ are orthogonal matrices ($F^T F = I, G^T G = I$), and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_p)$ is a diagonal matrix.

The matrices F and G may also be obtained from the eigen-decomposition (EVD) of A^TA and AA^T, respectively. In this case, $AA^T = F\Sigma^2F^T$ and $A^TA = G\Sigma^2G^T$. However, the matrices A^TA and AA^T are in general numerically ill-conditioned and thus the operation of computing the SVD of A from the EVD of A^TA or AA^T is numerically unstable and should be avoided.

The principal left singular subspace (L-PSS) of dimension r is the subspace spanned by the left singular vectors corresponding to the largest r singular values. Similarly, the principal right singular subspace (R-PSS) of dimension r is the subspace spanned by the right singular vectors corresponding to the largest r singular values. Methods for extracting principal singular subspace are proposed in [10]. Alternating power method and Bi-iteration SVD for tracking singular subspaces are developed in [14]-[15]. In this section, gradient flows for extracting the principal singular component analysis (PSCA) are derived.

4.1 Unconstrained Optimization

The principal singular subspaces can be obtained by maximizing the following Rayleigh quotient like function:

$$F_3(U,V) = tr\{(U^TU)^{\frac{-1}{2}}U^TAV(V^TV)^{\frac{-1}{2}}.\} \tag{28a}$$

To avoid the algebra and calculus of matrix square root, we consider the optimization problem $G_3(U,V) = tr(F_3(U,V)F_3(U,V)^T)$, where

$$G_3(U,V) = tr\{(U^TU)^{-1}U^TAV(V^TV)^{-1}V^TA^TU\}. \tag{28b}$$

It can be easily seen that $G_3(U,V) = G_3(UP,VQ)$ for any nonsingular matrices P and Q. Therefore, maximizing $G_3(U,V)$ or $F_3(U,V)$ will only produce an arbitrary basis of the principal singular subspace. Since our main purpose in this work is to develop algorithms for computing the actual left and right singular vectors, the cost functions of (28a) and (28b) are modified by incorporating a diagonal matrix D. Thus the goal is to maximize $G_4(U,V) = tr(F_3(U,V)D)$, where

$$G_4(U,V) = tr\{(U^TU)^{-1}U^TAV(V^TV)^{-1}V^TA^TUD\}. \tag{29}$$

It will be assumed that $D = \mathrm{diag}(d_1,\cdots,d_r)$ and that $d_1 > d_2 > \cdots > d_r > 0$.

For computational convenience and to avoid lengthy calculation of matrix square root derivatives, we will consider the modified doubly weighted Rayleigh quotient expression:

$$G_5(U,V) = tr\{(U^TU)^{-1}U^TAVD_1(V^TV)^{-1}V^TA^TUD_2\}. \tag{30}$$

Note that this function satisfies $G_5(UD_3,VD_4) = G_5(U,V)$ for any nonsingular diagonal matrices D_3, D_4.

The critical points of G_5 are solutions of $\nabla G_5 = 0$, where

$$\begin{aligned}\nabla_U G_5 &= AVD_1(V^TV)^{-1}V^TA^TUD_2(U^TU)^{-1}\\ &- U(U^TU)^{-1}U^TAVD_1(V^TV)^{-1}V^TA^TUD_2(U^TU)^{-1}\\ &- U(U^TU)^{-1}D_2U^TAV(V^TV)^{-1}V^TA^TUD_2(U^TU)^{-1}\\ &+ AV(V^TV)^{-1}D_1V^TA^TU(U^TU)^{-1}D_2,\end{aligned} \tag{31a}$$

and

$$\begin{aligned}\nabla_V G_5 &= A^TUD_2(U^TU)^{-1}U^TAVD_1(V^TV)^{-1}\\ &- V(V^TV)^{-1}V^TA^TUD_2(U^TU)^{-1}U^TAVD_1(V^TV)^{-1}\\ &+ A^TU(U^TU)^{-1}D_2U^TAV(V^TV)^{-1}D_1\\ &- V(V^TV)^{-1}D_1V^TA^TU(U^TU)^{-1}U^TAVD_1(V^TV)^{-1}.\end{aligned} \tag{31b}$$

If it is assumed that $U^TU = I$ and $V^TV = I$, then the above gradients reduce to

$$\begin{aligned}\nabla_U G_5 &= 2AVD_1V^TA^TUD_2 - U\{U^TAVD_1V^TA^TUD_2\\ &\quad + D_2U^TAVD_1V^TA^TU\},\\ \nabla_V G_5 &= 2A^TUD_2U^TAVD_1 - V\{V^TA^TUD_2U^TAVD_1\\ &\quad + D_1V^TA^TUD_2U^TAV\}.\end{aligned} \tag{32}$$

If the cost function

$$G_5(U,V) = tr\{(U^TU)^{-1}U^TAVD_1(V^TV)^{-1}V^TA^TUD_2\}. \tag{33}$$

is considered, then the following gradient flows are obtained

$$\begin{aligned}U' &= 2AVD_1V^TA^TUD_2D - U\{D_2U^TAVD_1V^TA^TUDD_2\\ &\quad + DD_2U^TAVD_1V^TA^TUD_2\},\\ V' &= 2A^TUDD_2U^TAVD_1 - V\{D_1V^TA^TUDD_2U^TAVD_1\\ &\quad + D_1V^TA^TUD_2DU^TAVD_1\}.\end{aligned} \tag{34}$$

Numerical experiments indicated that the gradient descent rules based on (32) and (34) are very slow convergent and are computationally demanding. Thus, in the next section we consider constrained problems with symmetric constraints so that the techniqies of Section 2 can be applied.

4.2 Constrained Optimization

For convenience of analysis we define the following set:

$$\Omega = \{(U,V) : U \in \mathbb{R}^{n\times p}, V \in \mathbb{R}^{m\times p} : U^TU > 0, V^TV > 0\}, \tag{35}$$

where the notation $X > 0$ stands for X being positive definite. To derive algorithms for true principal singular component analysis, the following optimization problem will be considered:

$$\mathrm{Maximize}_{(U,V)\in\Omega}\ tr\{G_6(U,V) = (U^TU)^{\frac{-1}{2}}U^TAV(V^TV)^{\frac{-1}{2}}D.\} \tag{36}$$

It can be shown that maximizing $tr(G_6(U,V))$ of (36) can be transformed into the following constrained optimization problems:

$$\text{Maximize } tr(U^TAVD) \text{ subject to } U^TU = I,\ V^TV = I, \tag{37a}$$

$$\text{Maximize } tr(U^TAV) \text{ subject to } U^TU = D^2,\ V^TV = I, \tag{37b}$$

$$\text{Maximize } tr(U^TAV) \text{ subject to } U^TU = D,\ V^TV = D, \tag{37c}$$

$$\text{Maximize } tr(U^TAV) \text{ subject to } U^TU = I,\ V^TV = D^2, \tag{37d}$$

$$\begin{aligned}&\text{Maximize } tr(U^TAV)\\ &\text{subject to } U^TU = D_1,\ V^TV = D_2,\ D_1D_2 = D^2,\end{aligned} \tag{37e}$$

where I is an identity matrix, D_1, D_2 are diagonal matrices of appropriate dimension. By forming the Lagrangian of each of these problems and applying the techniques of Section 2, we obtain Algorithms 1-6 listed in Section 4.4. Other learning rules may be obtained using the cost function:

$$\begin{aligned}&\text{Maximize } tr(U^TAVD_1V^TA^TUD_2)\\ &\text{subject to } U^TU = D_2^{-1},\ V^TV = D_1^{-1}.\end{aligned} \tag{38a}$$

This yields the following PSCA flow:

$$\begin{aligned}U' &= AVD_1V^TA^TUD_2 - UD_2U^TAVD_1V^TA^TUD_2\\ V' &= A^TUU^TAVD_1 - VD_1V^TA^TUU^TAVD_1.\end{aligned} \tag{38b}$$

Similarly, the differential equations associated with the optimization problem

$$\begin{aligned}&\text{Maximize } tr(U^TAVDV^TA^TU)\\ &\text{subject to } U^TU = D_1^{-1},\ V^TV = D_2^{-1},\end{aligned} \tag{39a}$$

are

$$\begin{aligned}U' &= AVDV^TA^TU - UU^TAVDV^TA^TUD_1\\ V' &= A^TUU^TAVD - VDV^TA^TUU^TAVD_2.\end{aligned} \tag{39b}$$

Several experiments were conducted to examine the performance of the SVD flows of (38b) and (39b). We noted that they are fast to converge to a permutation of the left and right principal singular vectors provided U and V are normalized at each iteration, however, it is very slow to converge otherwise.

4.3 Logarithmic Cost Functions

In addition to the Rayleigh quotient approaches, SVD algorithms can also be developed using logarithmic cost functions. We will consider the following optimization problems:

$$\mathrm{Maximize}_\Omega\ G_7(U,V) = tr\{\log(U^TAV+D) - U^TU - V^TV,\} \tag{40}$$

where D is a positive definite diagonal matrix. An algorithm which is based on the gradient flow of $G_7(U,V)$ is given in Algorithm 7 (see Section 4.4). A power method can also be developed as given in Algorithm 10 below.

4.4 Learning SVD Algorithms

Below is a list of several diagonal SVD algorithms. These are obtained by solving various optimization problems as indicated in the previous sections. Although numerous experiments were conducted to evaluate their performance, further analytical and experimental studies are needed to fully explain their convergence behavior. For most of the cost functions involved, we computed the Hessian matrix (not shown here) evaluated at desired critical points and they seem to be negative definite or negative semidefinite. All algorithms in this list appear to be new, although some of them reduce to known algorithms if D is an identity matrix or D is a null matrix. The parameter α is a positive learning rate and is usually a function of A and the initial matrices U_0 and V_0. The initial matrices U_0 and V_0 are assumed to be given and are full rank. A detailed analysis of these algorithms will be reported in a separate paper.

Algorithm 1

$$U_{k+1} = U_k + \alpha(AV_kD - U_kDV_k^TA^TU_k)$$
$$V_{k+1} = V_k + \alpha(A^TU_kD - V_kDU_k^TAV_k)$$

Algorithm 2

$$U_{k+1} = U_k + \alpha(AV_k - U_kV_k^TA^TU_kD^2)$$
$$V_{k+1} = V_k + \alpha(A^TU_k - V_kU_k^TAV_k)$$

Algorithm 3

$$U_{k+1} = U_k + \alpha(AV_k - U_kV_k^TA^TU_k)$$
$$V_{k+1} = V_k + \alpha(A^TU_k - V_kU_k^TAV_kD^2)$$

Algorithm 4

$$U_{k+1} = U_k + \alpha(AV_k - U_kV_k^TA^TU_kD)$$
$$V_{k+1} = V_k + \alpha(A^TU_k - V_kU_k^TAV_kD)$$

Algorithm 5

$$U_{k+1} = U_k + \alpha(AV_k - U_kV_k^TA^TU_kD_1),\ D_i : \text{diagonal}$$
$$V_{k+1} = V_k + \alpha(A^TU_k - V_kU_k^TAV_kD_2),\ D_1D_2 = D$$

Algorithm 6

$$U_{k+1} = U_k + \alpha(AV_kD_1 - U_kV_k^TA^TU_kD_2)$$
$$V_{k+1} = V_k + \alpha(A^TU_kD_3 - V_kU_k^TAV_kD_4),$$
provided that $D_1D_2^{-1}$ or $D_3D_4^{-1}$ has distinct eigenvalues.

Algorithm 7

$$U_{k+1} = U_k + \alpha(AV_k(U_k^TAV_k + D)^{-1} - U_k)$$
$$V_{k+1} = V_k + \alpha(A^TU_k(V_k^TAU_k + D)^{-1} - V_k)$$

Algorithm 8

$$U_{k+1} = U_k + \alpha(AVD(U_k^TAV_k)^{-1} - U_k)$$
$$V_{k+1} = V_k + \alpha(A^TU_kD(V_k^TA^TU_k)^{-1} - V_k)$$

Algorithm 9

$$U_{k+1} = U_k + \alpha(AV_kTri((U_k^TAV_k)^{-1}) - U_k)$$
$$V_{k+1} = V_k + a(A^TU_kTri((V_k^TA^TU_k)^{-1}) - V_k)$$

Here the notation $Tri(X)$ represents the upper or lower triangular part of X.

Algorithm 10

$$U_{k+1} = AV_k(U_k^TAV_k + D)^{-1}$$
$$V_{k+1} = A^TU_k(V_k^TA^TU_k + D)^{-1}$$

Algorithm 11

$$U_{k+1} = AV_kTri((U_k^TAV_k)^{-1})$$
$$V_{k+1} = A^TU_kTri((V_k^TA^TU_k)^{-1})$$

Algorithm 12

$$U_{k+1} = U_k + \alpha(AV_kD - U_kD(U_k^TAV_k + V_k^TA^TU_k)$$
$$V_{k+1} = V_k + \alpha(A^TU_kD - V_kD(U_k^TAV_k + V_k^TA^TU_k)$$

Remark 1: We should note that Algorithm 10 and 11 have very attractive properties. They are power methods and are globally convergent starting from any full rank initial matrices provided that the first r largest singular values of A are greater that the remaining singular values of A. Since they only require inversion of an $r \times r$ matrix, they are very efficient for computing a few singular values of large scale matrices.

Remark 2: In [15], the following cost function is minimized

$$F(U,V) = tr(A - UV^T)^T(A - UV^T) \tag{41}$$

to obtain a low-rank approximation for a rectangular matrix A. By minimizing the above function with respect to U and V alternatingly, the following algorithm is obtained:

$$\begin{aligned} U(k+1) &= AV(k)(V(k)^TV(k))^{-1} \\ V(k+1) &= A^TU(k)(U(k)^TU(k))^{-1} \end{aligned} \tag{42}$$

This is an alternating power method which converge fast to a solution (U,V) that is not unique and also is dependent on the initial matrices. Thus the alternating power method of (42) only produces an arbitrary basis of the principal singular subspace. It turns out that a slight modification of (42) motivated by the derivation of Algorithm 10, the alternating power method could produce the actual low rank SVD. Thus the new alternating power method for SVD is given as in the following algorithm.

Algorithm 13

$$\begin{aligned} U(k+1) &= AV(k)(V(k)^TV(k) + D)^{-1} \\ V(k+1) &= A^TU(k)(U(k)^TU(k) + D)^{-1}, \end{aligned}$$

where D is a diagonal matrix with positive diagonal entries. This algorithm computes the true singular value components in that it generates a sequence $(U(k),V(k))$ such that $U(k)^TU(k)$, $V(k)^TV(k)$, and $U(k)^TAV(k)$ converge to diagonal matrices as $k \to \infty$. To show this, let $(U(k),V(k)) \to (U_\infty, V_\infty)$, then $U_\infty^TU_\infty(U_\infty^TU_\infty + D) = U_\infty^TAV_\infty$ and $V_\infty^TV_\infty(V_\infty^TV_\infty + D) = V_\infty^TA^TU_\infty$. Clearly,

$$\begin{aligned} D\{(U_\infty^TU_\infty)^2 - U_\infty^TAV_\infty\} &= \{(U_\infty^TU_\infty)^2 - V_\infty^TA^TU_\infty\}D \\ D\{(V_\infty^TV_\infty)^2 - V_\infty^TA^TU_\infty\} &= \{(V_\infty^TV_\infty)^2 - U_\infty^TAV_\infty\}D. \end{aligned} \tag{43}$$

This shows that

$$\begin{aligned} &D((U_\infty^TU_\infty)^2 - U_\infty^TAV_\infty + (V_\infty^TV_\infty)^2 - V_\infty^TA^TU_\infty) \\ &= ((U_\infty^TU_\infty)^2 - V_\infty^TA^TU_\infty + (U_\infty^TU_\infty)^2 - U_\infty^TAV_\infty)D \end{aligned} \tag{44}$$

Hence, $(U_\infty^TU_\infty)^2 - U_\infty^TAV_\infty + (V_\infty^TV_\infty)^2 - V_\infty^TA^TU_\infty = D_2$ for some diagonal matrix D_2. It follows from Proposition 4 (see Appendix) that $(V_\infty^TV_\infty)D = (V_\infty^TV_\infty)^2 - V_\infty^TA^TU_\infty$ and $(U_\infty^TU_\infty)D = (U_\infty^TU_\infty)^2 - U_\infty^TAV_\infty$ are diagonal matrices. Hence $V_\infty^TV_\infty$ and $U_\infty^TU_\infty$ are diagonal. This implies that $U_\infty^TAV_\infty = U_\infty^TU_\infty(U_\infty^TU_\infty + D)$ and $V_\infty^TA^TU_\infty = V_\infty^TV_\infty(V_\infty^TV_\infty + D)$ are diagonal as they are products of diagonal matrices.

Motivated by Algorithm 11, another modification of Algorithm 13 can be obtained:

Algorithm 14

$$U_{k+1} = AV_kTri((V_k^TV_k)^{-1})$$
$$V_{k+1} = A^TU_kTri((U_k^TU_k)^{-1})$$

Here the notation $Tri(X)$ represents the upper or lower triangular part of X. Simulations showed that $U_k^TU_k$,

$V_k^T V_k$ and $U_k^T A V_k$ converge to diagonal matrices as $k \rightarrow \infty$. Thus the PSCA of dimension r will be of the form: $F_r = U_\infty (U_\infty^T U_\infty)^{\frac{-1}{2}} P$, $G_r = V_\infty (V_\infty^T V_\infty)^{\frac{-1}{2}} P$, and $\Sigma_r = P^T (U_\infty^T U_\infty)^{\frac{-1}{2}} U_\infty^T A V_\infty (V_\infty^T V_\infty)^{\frac{-1}{2}} P$, where P is a permutation matrix. We noted that Algorithm 11 is superior to Algorithm 14 in that U_k and V_k are automatically normalized in Algorithm 11. One may also argue that Algorithm 11 is simpler and easier to implement that the Bi-iteration proposed in [14] as no QR factorization is required.

5. Proofs

In the last few sections, we stated several results and learning algorithms for PCA, MCA, and SVD. The derivation of these results and their proofs follow staraightforward from applying the methodology described in Section 2. Due to space limitations, we will only provide a proposition for the derivation of the critical points and critical values of the optimization problem (37a) which yields Algorithm 1. Other optimization problems of (37b)-(37f) can be treated similarly. The gradient of matrix functions is computed by applying the matrix calculus presented in [16].

Proposition 3. *Let D be a diagonal matrix such that the diagonal entries of D are positive, distinct, and arranged in descending order and let $A \in \mathbb{R}^{m \times n}$ be a real matrix with singular values $\sigma_1 > \cdots > \sigma_r > \sigma_{r+1} \geq \cdots \geq \sigma_p \geq 0$ and the corresponding orthonormal left and right singular vectors are $F = [f_1, \cdots, f_p]$ and $G = [g_1, \cdots, g_p$, respectively. Let $G_8(U,V) = tr(U^T A V D)$, then the maximum of $G_8(U,V)$, subject to $U^T U = I$, and $V^T V = I$ is attained if and only if $U = F_r$ and $V = G_r$, where $F_r = [f_1 \cdots f_r]$ and $G_r = [g_1 \cdots g_r]$. Moreover, $\max_\Omega tr(U^T A V D) = \sum_{i=1}^r \sigma_i d_i$.*

Outline of Proof: Let the Lagrangian of the optimization problem (37a) be

$$\mathcal{L}(U,V) = tr\{U^T A V D - (U^T U - I)\frac{\lambda_1}{2} - (V^T V - I)\frac{\lambda_2}{2}\},$$

where λ_1 and λ_2 are Lagrange multiplier matrices. Since this is an optimization problem over a compact set, both minima and maxima exist. For any critical point (U,V) of (37a), $\nabla\mathcal{L}(U,V) = 0$, where

$$\nabla\mathcal{L}(U,V) = \begin{bmatrix} AVD - U\lambda_1 \\ A^T U D - V\lambda_2 \\ U^T U - I \\ V^T V - I \end{bmatrix}.$$

For any two matrices U and V satisfying $U^T U = I$ and $V^T V = I$, the Lagrange multipliers λ_1 and λ_2 are given as: $\lambda_1 = U^T \nabla_U \mathcal{L} = U^T A V D$ and $\lambda_2 = V^T \nabla_V \mathcal{L} = V^T A^T U D$. Additionally, both λ_1 and λ_2 are symmetric. Clearly, (U,V) is a solution of $\nabla\mathcal{L}(U,V) = 0$ if U is a matrix whose columns consists of linear combination of r left singular vectors, and V is the matrix each column of which is a linear combination of corresponding right singular vectors, i.e., $U = \bar{F}_r P$ and $V = \bar{G}_r Q$, where P and Q are orthogonal matrices. Here $\bar{F}_r$ is a matrix consisting of any r columns of F and $\bar{G}_r$ is a matrix consisting of the corresponding r columns of G. Now, for any critical point (U,V), it follows that $tr(U^T A V D) = tr(P^T \bar{\Sigma}_r Q D)$, where $\bar{\Sigma}_r$ is a diagonal matrix so that $\bar{F}_r^T A \bar{G}_r = \bar{\Sigma}_r$. From the necessary condition for optimality, we get $\bar{\Sigma}_r Q D - P\lambda_1 = 0$, and $\bar{\Sigma}_r P D - Q\lambda_2 = 0$. Since $\lambda_1^T = \lambda_1$ and $\lambda_2^T = \lambda_2$, it follows that $\lambda_1 = P^T \bar{\Sigma}_r Q D = D Q^T \bar{\Sigma}_r P$, and $\lambda_2 = Q^T \bar{\Sigma}_r P D = D P^T P^T \bar{\Sigma}_r Q$. Let $E = P^T \bar{\Sigma}_r Q$, then $ED = DE^T$ and $E^T D = DE$. This implies that $(E + E^T)D = D(E + E^T)$, and therefore Proposition 4 (see Appendix), $(E + E^T) = D_1$ for some diagonal matrix D_1. Now, $ED = DE^T = D(D_1 - E)$, or $ED + DE = DD_1$. Proposition 4 (see Appendix) guarantees that E is diagonal. Assume that $E = P^T \bar{\Sigma}_r Q = D_2$, then $Q = \bar{\Sigma}_r^{-1} P D_2$ and hence, $I = Q^T Q = D_2 P^T \bar{\Sigma}_r^{-2} P D_2$. This shows that $P^T \bar{\Sigma}_r^{-2} P = D_2^{-2}$. The last equation implies that P is a permutation matrix. Similarly, one can show that Q is a permutation matrix. The equation $E = P^T \bar{\Sigma}_r Q = D_2$ implies that $Q^T P = D_2^{-1} P^T \bar{\Sigma}_r^{-1} P = D_3$ for some diagonal matrix D_3. Since P and Q are permutation matrices, we must have $P = Q$. The value of the cost function $U^T A V D$ at a critical point has the form $tr(P^T \bar{\Sigma}_r P D) = \sum_{i=1}^r \sigma_{j_i} d_i$. Since $\bar{\Sigma}_r$ and D have positive diagonal element (in decreasing order), the maximum of $tr(P^T \bar{\Sigma}_r Q D)$ occurs when $P = I$, in which case the maximum is $\sum_{i=1}^r \sigma_i d_i$. It also follows that the maximum is attained at $U = F_r$ and $V = G_r$. Q.E.D.

Appendix: Finally, we state a result which is essential for the proofs of Propositions 1-3.

Proposition 4 [17]. *Let $D, C \in \mathbb{R}^{n \times n}$ such that D is diagonal having distinct eigenvalues. If $CD = DC$, then C is diagonal.*

References

[1] G. Cirrincione, M. Cirrincione, J. Herault, and S. Van Huffel, "The MCA EXIN Neuron for the Minor Component Analysis," IEEE Trans. on Neural Networks, Vol. 13, No. 1, pp. 160-187, January 2002.

[2] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward network," Neural Networks, 2:459-473, 1989.

[3] E. Oja, "Principal components, minor components, and linear neural networks. Neural Networks," 5:927-935, November 1992.

[4] L. Xu, "Least mean square error recognition principle for self organizing neural nets," Neural Networks, 6:627-648, 1993.

[5] T. Chen and Shun-ichi Amari, "Unified Stabilization Approach to Principal and Minor Components Algorithms," Neural Networks, Vol. 14, pp. 1377-1387, 2001.

[6] Yongfeng Miao; Yingbo Hua, "Fast subspace tracking by a novel information criterion," Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers, 1997, Volume: 2, pp.1312 - 1316, 2-5 Nov. 1997.

[7] Hori, G., "A general framework for SVD flows and joint SVD flows," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003 (ICASSP '03), Volume: 2, 6-10, April 2003, pp:II-693-696.

[8] J. B. Moore, R. E. Mahony, and U. Helmke, "Numerical gradient algorithms for eigenvalues and singular value calculations," SIAM J. Matrix Anal. Appl., vol. 15, pp. 881902, 1994.

[9] A. Cichocki, Neural network for singular value decomposition, Electron. Lett., vol. 28, pp. 784786, 1992.

[10] Da-Zheng Feng; Xian-Da Zhang; Zheng Bao, "A neural network learning for adaptively extracting cross-correlation features between two high-dimensional data streams," IEEE Transactions on Neural Networks, Volume: 15, Issue: 6, pp. 1541-1554, Nov. 2004.

[11] A. Edelman, T. A. Arias and S. T. Smith, "The geometry of algorithms with orthogonality constraints" SIAM J. Matrix Anal. Appl., 20(2):303-353, 1998.

[12] J.H. Manton, "Optimisation algorithms exploiting unitary constraints," IEEE Trans. Signal Processing, vol. 50, pp. 635-650, 2002.

[13] Shan Ouyang; Zheng Bao; Gui-Sheng Liao; Ching, P.C., "Adaptive minor component extraction with modular structure," IEEE Transactions on Signal Processing, Volume: 49, Issue: 9, pp:2127- 2137, Sept. 2001.

[14] P. Strobach, "Bi-iteration SVD subspace tracking algorithms," IEEE Trans. Signal Processing, vol.45, no.5, pp.1222-1240, 1997.

[15] Shan Ouyang; Yingbo Hua, "Bi-iterative least square versus bi-iterative singular value decomposition for subspace tracking," Proceedings, Acoustics, Speech, and Signal Processing, 2004 (ICASSP '04), Volume: 2, 17-21 May 2004, pp- 353-356.

[16] J. R. Magnus and H. Neudecker, Matrix Differential Calculus with Applications in Statistics and Econometrics, 2nd ed. New York: Wiley, 1991.

[17] Hasan, M.A., "Natural Gradient for Minor Component Extraction," to be presented at the 2005 International Symposium on Circuits and Systems, May 2005.