Incremental Eigen Decomposition

James T. Kwok and Haitao Zhao
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay
Hong Kong
Email: [jamesk,zhaoht]@cs.ust.hk

***Abstract*— Eigen decomposition is a central mathematical tool in many pattern recognition and machine learning techniques. However, it becomes computationally infeasible in the presence of a large set of high-dimensional samples. Moreover, in highly dynamic domains with a continual supply of new samples, an incremental approach that keeps on updating the eigen decomposition will be more desirable than the traditional approach of simply re-computing the decomposition from scratch. In this paper, by using a method for updating singular value decompositions, we propose a procedure to obtain an approximate eigen decomposition by processing the samples in chunks or sequentially. On applying this to principal component analysis for image denoising, experimental results show that this restricted form of updating can still achieve comparable performance as the batch version.**

I. Introduction

Eigen decomposition is a central mathematical tool in many pattern recognition and machine learning techniques, such as principal component analysis (PCA), Fisher discriminant analysis and, more recently, spectral methods like locally linear embedding. Take PCA as an example, eigen decomposition is to be operated on the $d \times d$ sample covariance matrix (where d is the input dimensionality), typically using numerical methods like the power method or QL algorithm [2]. However, this requires batching all the samples together before performing PCA, which may not be practical in highly dynamic domains with a continual supply of new samples. Another problem is that the eigen decomposition in PCA will be computationally infeasible when d becomes large. A trick, as is often employed in implementing eigenfaces [7], is to operate instead on the $n \times n$ inner-product matrix (where n is the number of samples), since $\mathbf{X}\mathbf{X}'$ and $\mathbf{X}'\mathbf{X}$ share the same set of nonzero eigenvalues for any matrix $\mathbf{X}$. However, in situations where both n and d are large (as in processing a large image database), performing eigen decomposition on either matrix will still be infeasible. One remedy is to subsample the data, which inevitably incurs some loss of information. An alternative is to adopt an incremental approach, in which samples are processed in chunks or sequentially. In [4], [5], [8], this is accomplished by avoiding estimating the covariance matrix all together. However, they may suffer from the problem of slow convergence.

Another numerical method for obtaining eigen decomposition is by singular value decomposition (SVD). In particular, if the SVD of matrix $\mathbf{X}$ is $\mathbf{U}\Sigma\mathbf{V}'$. Then, $\mathbf{X}'\mathbf{X} = \mathbf{V}\Sigma\mathbf{U}'\mathbf{U}\Sigma\mathbf{V}' = \mathbf{V}\Sigma^2\mathbf{V}'$, which is the eigen decomposition of $\mathbf{X}'\mathbf{X}$. In other words, the eigenvalues of $\mathbf{X}'\mathbf{X}$ are the squared singular values of $\mathbf{X}$ and the eigenvectors of $\mathbf{X}'\mathbf{X}$ are the right singular vectors of $\mathbf{X}$. Besides, the best rank-k approximation of $\mathbf{X}$ will retain the k largest singular values of $\mathbf{X}$, which correspond to the k largest eigenvalues of $\mathbf{X}'\mathbf{X}$ [2].

Recently, in the information retrieval community, proposals have been made to update SVD sequentially with the arrival of more samples. Examples include [1] (which is commonly called *folding-in*), [3] and the more accurate [9]. In this paper, we will adapt [9] for the updating of eigen decompositions. This can then be used to alleviate the problems associated with batch PCA as mentioned above.

The rest of this paper is organized as follows. Brief review on the SVD updating procedure in [9] is given in Section II. Section III describes our proposed eigen decomposition updating method. Section IV presents experimental results on applying this to PCA-based image denoising, and the last section gives some concluding remarks.

II. Updating SVD

Given $\mathbf{A}_{m\times n}$ and its best rank-k approximation $\hat{\mathbf{A}}_{m\times n} = \mathbf{U}\Sigma\mathbf{V}'$, we review the procedure to obtain the best rank-k approximations of $[\hat{\mathbf{A}}_{m\times n}, \mathbf{B}_{m\times r}]$ and $\begin{bmatrix} \hat{\mathbf{A}}_{m\times n} \\ \mathbf{B}_{r\times n} \end{bmatrix}$ as described in [9]. Typically, approximations of $[\mathbf{A}, \mathbf{B}]$ and $[\mathbf{A}', \mathbf{B}']'$ will be of more interest. However, they are also more difficult to obtain. Moreover, the rank-k approximations of $[\hat{\mathbf{A}}, \mathbf{B}]$ and $[\hat{\mathbf{A}}', \mathbf{B}']'$ will be close to those of $[\mathbf{A}, \mathbf{B}]$ and $[\mathbf{A}', \mathbf{B}']'$ when these matrices have the so-called *low-rank-plus-shift* structure [9], [10].

A. Adding Columns

We first review the procedure on obtaining the best rank-k approximation of $[\hat{\mathbf{A}}, \mathbf{B}]$.

1) Obtain the QR decomposition $(\mathbf{I} - \mathbf{U}\mathbf{U}')\mathbf{B} = \mathbf{Q}\mathbf{R}$.
2) Obtain the rank-k SVD of the $(k + r') \times (k + r)$ matrix $\begin{bmatrix} \Sigma & \mathbf{U}'\mathbf{B} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} = \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}'$, where r' is the rank of $(\mathbf{I} - \mathbf{U}\mathbf{U}')\mathbf{B}$.
3) The best rank-k approximation of $[\hat{\mathbf{A}}, \mathbf{B}]$ is then $\left([\mathbf{U}, \mathbf{Q}]\tilde{\mathbf{U}}\right) \tilde{\Sigma} \left(\begin{bmatrix} \mathbf{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tilde{\mathbf{V}} \right)'$.

B. Adding Rows

The procedure for obtaining the best rank-k approximation of $[\hat{\mathbf{A}}', \mathbf{B}']'$ is analogous.

1) Obtain the QL decomposition $(\mathbf{I} - \mathbf{V}\mathbf{V}')\mathbf{B}' = \mathbf{Q}\mathbf{L}'$.
2) Obtain the rank-k SVD of the $(k+r)\times(k+r')$ matrix $\begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{BV} & \mathbf{L} \end{bmatrix} = \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}'$, where r' is the rank of $(\mathbf{I} - \mathbf{V}\mathbf{V}')\mathbf{B}'$.
3) The best rank-k approximation of $[\hat{\mathbf{A}}', \mathbf{B}']'$ is $\left(\begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}\tilde{\mathbf{U}}\right)'\tilde{\Sigma}\left([\mathbf{V},\mathbf{Q}]\tilde{\mathbf{V}}\right)$.

III. Updating Eigen Decomposition

Given the rank-k eigen decomposition of an $m\times m$ positive definite matrix $\mathbf{C}_1$, we now consider the rank-k eigen decomposition of an expanded $(m+r)\times(m+r)$ positive definite matrix $\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 \\ \mathbf{C}_2' & \mathbf{C}_3 \end{bmatrix}$, where $\mathbf{C}_2$ is of size $m\times r$ and $\mathbf{C}_3$ is of size $r\times r$.

A. Decomposing $\mathbf{C}$ as $\mathbf{A}'\mathbf{A}$

As mentioned in Section I, the SVD of $\mathbf{A}$ is related to the eigen decomposition of $\mathbf{A}'\mathbf{A}$. In this Section, we first attempt to decompose the expanded $\mathbf{C}$ matrix into this form.

Suppose that the eigen decomposition[1] of $\mathbf{C}_1$ is $\mathbf{C}_1 = \mathbf{U}\Lambda\mathbf{U}' = (\Lambda^{\frac{1}{2}}\mathbf{U}')'(\Lambda^{\frac{1}{2}}\mathbf{U}') = \mathbf{A}'\mathbf{A}$, where $\mathbf{A}_{m\times m} = \Lambda^{\frac{1}{2}}\mathbf{U}'$. Consider decomposing $\mathbf{C}$ as

$$\mathbf{C} = \begin{bmatrix} \mathbf{A} & \mathbf{B}_1 \\ \mathbf{B}_2 & \mathbf{B}_3 \end{bmatrix}'\begin{bmatrix} \mathbf{A} & \mathbf{B}_1 \\ \mathbf{B}_2 & \mathbf{B}_3 \end{bmatrix},$$

where $\mathbf{B}_1$ is $m\times r$, $\mathbf{B}_2$ is $r\times m$, and $\mathbf{B}_3$ is $r\times r$. We proceed to show how these $\mathbf{B}_1, \mathbf{B}_2$ and $\mathbf{B}_3$ can be obtained.

Now,

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 \\ \mathbf{C}_2' & \mathbf{C}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{A}'\mathbf{A} & \mathbf{C}_2 \\ \mathbf{C}_2' & \mathbf{C}_3 \end{bmatrix}, \tag{1}$$

and

$$\begin{aligned} &\begin{bmatrix} \mathbf{A} & \mathbf{B}_1 \\ \mathbf{B}_2 & \mathbf{B}_3 \end{bmatrix}'\begin{bmatrix} \mathbf{A} & \mathbf{B}_1 \\ \mathbf{B}_2 & \mathbf{B}_3 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}' & \mathbf{B}_2' \\ \mathbf{B}_1' & \mathbf{B}_3' \end{bmatrix}\begin{bmatrix} \mathbf{A} & \mathbf{B}_1 \\ \mathbf{B}_2 & \mathbf{B}_3 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}'\mathbf{A} + \mathbf{B}_2'\mathbf{B}_2 & \mathbf{A}'\mathbf{B}_1 + \mathbf{B}_2'\mathbf{B}_3 \\ \mathbf{B}_1'\mathbf{A} + \mathbf{B}_3'\mathbf{B}_2 & \mathbf{B}_1'\mathbf{B}_1 + \mathbf{B}_3'\mathbf{B}_3 \end{bmatrix}. \end{aligned} \tag{2}$$

Equating terms corresponding to $\mathbf{C}_1$ in (1) and (2), we have $\mathbf{B}_2'\mathbf{B}_2 = \mathbf{0}$ and hence $\mathbf{B}_2 = \mathbf{0}$. (2) then simplifies to

$$\begin{bmatrix} \mathbf{A}'\mathbf{A} & \mathbf{A}'\mathbf{B}_1 \\ (\mathbf{A}'\mathbf{B}_1)' & \mathbf{B}_1'\mathbf{B}_1 + \mathbf{B}_3'\mathbf{B}_3 \end{bmatrix}.$$

On equating terms for $\mathbf{C}_2$, we have $\mathbf{A}'\mathbf{B}_1 = \mathbf{C}_2$. As $\mathbf{C}_1$ is positive definite, it is invertible and so is Λ. Hence,

$$\mathbf{B}_1 = (\mathbf{A}')^{-1}\mathbf{C}_2 = \Lambda^{-\frac{1}{2}}\mathbf{U}'\mathbf{C}_2, \tag{3}$$

[1] Notice that this is the complete eigen decomposition (not the rank-k approximation) and Λ contains all the eigenvalues (some of them may be small) of $\mathbf{C}_1$.

Finally, equating terms for $\mathbf{C}_3$, we have

$$\mathbf{B}_3'\mathbf{B}_3 = \mathbf{C}_3 - \mathbf{B}_1'\mathbf{B}_1, \tag{4}$$

and so $\mathbf{B}_3$ can be obtained as the square root of $\mathbf{C}_3 - \mathbf{B}_1'\mathbf{B}_1$, if $\mathbf{C}_3 - \mathbf{B}_1'\mathbf{B}_1$ is positive definite. To show that this is indeed the case, we first write

$$\begin{aligned} \mathbf{B}_1'\mathbf{B}_1 &= \mathbf{C}_2'\mathbf{U}\Lambda^{-\frac{1}{2}}\Lambda^{-\frac{1}{2}}\mathbf{U}'\mathbf{C}_2 = \mathbf{C}_2'\mathbf{U}\Lambda^{-1}\mathbf{U}'\mathbf{C}_2 \\ &= \mathbf{C}_2'\mathbf{C}_1^{-1}\mathbf{C}_2. \end{aligned}$$

Consider $\mathbf{P} = \begin{bmatrix} \mathbf{I} & -(\mathbf{C}_1^{-1})'\mathbf{C}_2 \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$, where $\mathbf{I}$ is the identity matrix of the appropriate size. Then, $\mathbf{P}'\mathbf{C}\mathbf{P} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_3 - \mathbf{C}_2'\mathbf{C}_1^{-1}\mathbf{C}_2 \end{bmatrix}$. As $\mathbf{P}$ is non-singular, $\mathbf{P}'\mathbf{C}\mathbf{P}$ is similar to $\mathbf{C}$ and so is also positive definite (as $\mathbf{C}$ is positive definite) and thus so is $\mathbf{C}_3 - \mathbf{C}_2'\mathbf{C}_1^{-1}\mathbf{C}_2 = \mathbf{C}_3 - \mathbf{B}_1'\mathbf{B}_1$. Hence, from (4), $\mathbf{B}_3$ can be obtained as the square root of $\mathbf{C}_3 - \mathbf{B}_1'\mathbf{B}_1$, an $r\times r$ matrix.

B. Procedure

The idea of the proposed incremental procedure is as follows. The rank-k eigen decomposition of $\mathbf{C}_1 = \mathbf{A}'\mathbf{A}$ corresponds to the best rank-k approximation $\hat{\mathbf{A}}^{(1)}$ of $\mathbf{A}$. Then, we can obtain the best rank-k approximation $\hat{\mathbf{A}}^{(2)}$ of $\begin{bmatrix} \hat{\mathbf{A}}^{(1)} \\ \mathbf{B}_2 \end{bmatrix}$, and subsequently that of $\left[\hat{\mathbf{A}}^{(2)}, \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_3 \end{bmatrix}\right]$ using the method in Section II. This will then serve as an approximation to the SVD of $\begin{bmatrix} \mathbf{A} & \mathbf{B}_1 \\ \mathbf{B}_2 & \mathbf{B}_3 \end{bmatrix}$, from which we can get back an approximate eigen decomposition of $\begin{bmatrix} \mathbf{A} & \mathbf{B}_1 \\ \mathbf{B}_2 & \mathbf{B}_3 \end{bmatrix}'\begin{bmatrix} \mathbf{A} & \mathbf{B}_1 \\ \mathbf{B}_2 & \mathbf{B}_3 \end{bmatrix} = \mathbf{C}$.

Denote the best rank-k approximation of $\mathbf{A}$ by $\hat{\mathbf{A}}^{(1)} = \mathbf{U}\Sigma_k\mathbf{V}'$. The following procedure describes the above idea in more detail:

1) Obtain $\mathbf{B}_1$ from (3) and $\mathbf{B}_3$ as the square root of $\mathbf{C}_3 - \mathbf{B}_1'\mathbf{B}_1$ in (4).
2) Recall that $\mathbf{B}_2 = \mathbf{0}$, thus $\begin{bmatrix} \hat{\mathbf{A}}^{(1)} \\ \mathbf{B}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{U}\Sigma_k\mathbf{V}' \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{U} \\ \mathbf{0} \end{bmatrix}\Sigma_k\mathbf{V}'$, which is of rank k already. Hence, the best rank-k approximation $\hat{\mathbf{A}}^{(2)}$ of $\begin{bmatrix} \hat{\mathbf{A}}^{(1)} \\ \mathbf{0} \end{bmatrix}$ is simply $\begin{bmatrix} \hat{\mathbf{A}}^{(1)} \\ \mathbf{0} \end{bmatrix}$ itself.
3) Obtain the best rank-k approximation $\hat{\mathbf{A}}^{(3)}$ of $\left[\hat{\mathbf{A}}^{(2)}, \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_3 \end{bmatrix}\right]$ using Section II-A.
4) Take the resulting squared singular values and right singular vectors as the eigenvalues and eigenvectors of $\mathbf{C}$.

Notice that although, conceptually, the procedure involves first the addition of rows and then columns, the fact that $\mathbf{B}_2 = \mathbf{0}$ means that there is no actual computation in Step 2 above. Thus, only one round of approximation based on column addition is required.

C. Remarks

When this incremental procedure is used for PCA, the matrix $\mathbf{C}$ will be the sample covariance matrix. However, a limitation is that the $\mathbf{C}_1$ sub-matrix cannot be changed when the covariance matrix is expanded. This implies that the sample mean is assumed to remain constant, which may not hold especially when a large number of samples are added. In such cases, we may instead perform non-centered PCA [2], [6], which does not require centering ($\mathbf{C}$ is then the outer-product matrix). Non-centered PCA has been commonly used in ecology and chemistry, and has similar behavior as the standard (centered) PCA in general.

IV. Experiments

In this Section, we compare the performance of using batch and incremental (non-centered) PCA for image denoising on the USPS dataset of 16×16 handwritten digits[2]. An initial set of 100 samples of the digit zero are chosen to form the covariance matrix $\mathbf{C}_1$. Samples are then added in increments of 25, up to a maximum total number of 250 samples. The test set consists of another 100 samples, which are corrupted by additive, zero-mean Gaussian noise of variance 0.1. Performance is measured by the mean squared error of the denoised images:

$$\frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{16^2}\sum_{i=1}^{16}\sum_{j=1}^{16}(I_{ij}-\hat{I}_{ij})^2\right),$$

where I is the clean test image, $\hat{I}$ is the denoised image and N (=100) is the number of test images. Notice that while batch PCA is possible on the small dataset here, it may not be feasible in general.

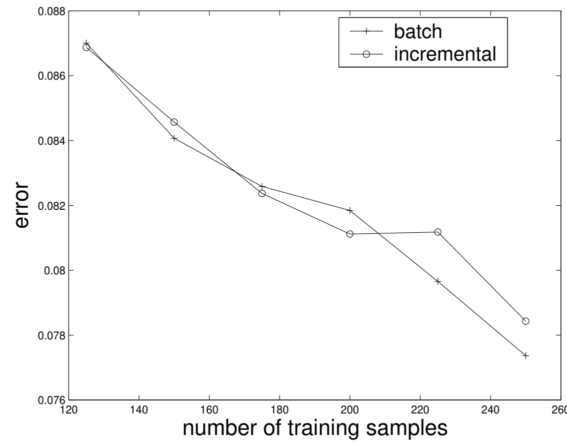
Figure 1 compares the performance when 30, 50 and 70 principal components are used in the eigen decomposition updating and denoising[3]. Notice that although the incremental procedure only computes an approximate eigen decomposition, its denoising performance is comparable and sometimes even outperforms that of batch PCA. Figures 2 and 3 compare the two sets of eigenvalues/eigenvectors for the setting with 50 principal components and 250 samples. Notice in particular that the two curves in Figure 2 have overlapped almost perfectly.

V. Conclusion

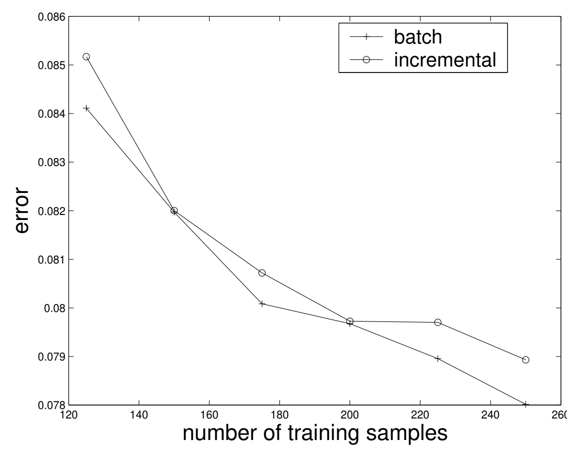
In this paper, we propose an incremental procedure for updating eigen decomposition. Experimental results on applying this to PCA-based image denoising show that it achieves comparable performance with standard, batch PCA. In the future, we will also apply this incremental procedure to other techniques that rely on eigen decomposition, such as Fisher discriminant analysis, spectral methods and their kernelized extensions.

[2]The USPS database can be downloaded from http://www.kernel-machines.org.

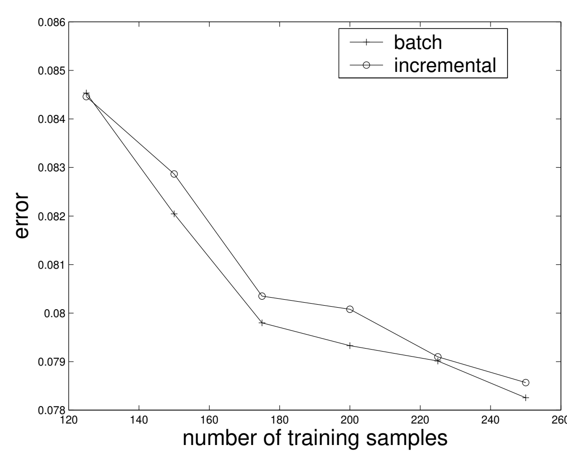
[3]Alternatively, one may keep a larger number of principal components for eigen decomposition updating, but use only a subset of the leading components for denoising.



(a) 30 PCs.



(b) 50 PCs.



(c) 70 PCs.

Fig. 1. Denoising errors at different numbers of principal components (PCs).

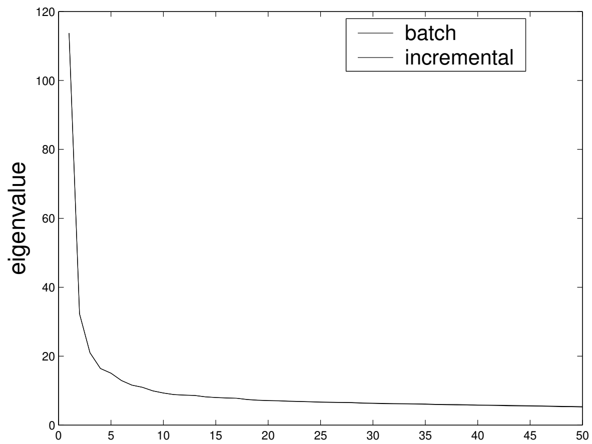


Fig. 2. Comparison of the eigenvalues.

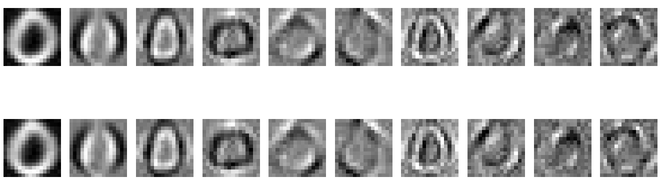


Fig. 3. Comparison of the first 10 eigenvectors. Top: batch PCA; bottom: incremental PCA.

ACKNOWLEDGMENTS

This research has been partially supported by the Research Grants Council of the Hong Kong Special Administrative Region under grants HKUST2033/00E and HKUST6195/02E.

REFERENCES

[1] M.W. Berry, S.T. Dumais, and G.W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, December 1995.

[2] I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, second edition, 2002.

[3] G. O'Brien. Information management tools for updating an SVD-encoded indexing scheme. Master's thesis, University of Tennessee, Knoxville, TN, 1994.

[4] E. Oja and J. Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of Mathematical Analysis and Applications*, 106:69–84, 1985.

[5] T.D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *IEEE Transactions on Neural Networks*, 2:459–473, 1989.

[6] C.J.F. Ter Braak. Principal components biplots and Alpha and Beta diversity. *Ecology*, 64(3):454–462, 1983.

[7] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[8] J. Weng, Y. Zhang, and W.-S. Hwang. Complementary candid incremental principal component analysis. To appear in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[9] H. Zha and H.D. Simon. On updating problems in latent semantic indexing. *SIAM Journal on Scientific Computing*, 21(2):782–791, 1999.

[10] H. Zha and Z. Zhang. On matrices with low-rank-plus-shift structures: Partial SVD and latent semantic indexing. *SIAM Journal of Matrix Analysis and Applications*, 21:522–536, 1999.