SEQUENTIAL EM LEARNING FOR SUBSPACE ANALYSIS

Seungjin Choi

Department of Computer Science & Engineering, POSTECH, Korea
seungjin@postech.ac.kr

ABSTRACT

Subspace analysis (which includes PCA) seeks for feature subspace (which corresponds to the eigenspace), given multivariate input data and has been widely used in computer vision and pattern recognition. Typically data space belongs to very high dimension, but only a few principal components need to be extracted. In this paper I present a fast sequential algorithm for subspace analysis or tracking. Useful behavior of the algorithm is confirmed by numerical experiments.

1. INTRODUCTION

The task of pattern recognition is to classify each data in an unlabelled set of data (test set), given a set of data labelled with its class (training set). Statistical pattern recognition relies on information coding (data representation). For example, linear data representation decomposes a given set of data into a linear sum of basis vectors. Feature vectors are obtained by projecting the data onto the subspace spanned by the basis vectors. Popular methods are factor analysis and principal component analysis (PCA). The eigenface method [8, 11] might be one exemplary subspace analysis technique in pattern recognition.

In computer vision and pattern recognition, one often encounters into a set of huge dimensional data and wish to extract a small number of features which is able to represent the data as well as possible. The singular value decomposition (SVD) is a numerically robust method which calculates the eigenvectors of the covariance matrix, however, it is computationally expensive, especially for the case of data with high dimension. For adaptive computation of eigenvectors, a variety of PCA neural networks have been developed [1], most of which are gradient-based learning algorithms, so their convergence is very slow.

Recently probabilistic model-based methods for subspace analysis have been proposed. These include probabilistic PCA (PPCA) [10], EM-PCA [5], mixtures of factor analyzers [2], and mixtures of probabilistic principal component analyzers [9]. All these algorithms are employ the EM learning which is an iterative maximum likelihood estimation method in the presence of hidden variables. PPCA and EM-PCA are batch algorithms, thus, when a new data arrives, whole calculation should be carried out again. In order to overcome this drawback, I present a sequential EM learning algorithm.

2. LINEAR GENERATIVE MODEL

The linear generative model assumes that the set of m dimensional observed vectors $\{\boldsymbol{x}_t\}$ is generated from a corresponding set of latent variables $\{\boldsymbol{s}_t\}$ by

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{s}_t + \boldsymbol{v}_t, \tag{1}$$

where $\boldsymbol{s}_t \in \mathbb{R}^n$ $(n \leq m)$ and $\boldsymbol{v}_t$ is Gaussian noise vector that is assumed to be statistically independent of $\boldsymbol{s}_t$.

In standard factor analysis, latent variables $\boldsymbol{s}$ are assumed to have a unit isotropic Gaussian distribution, i.e., $\boldsymbol{s} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The noise model is Gaussian, i.e., $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma}$ being a diagonal matrix. Given this formulation, the model for $\boldsymbol{x}$ is also Gaussian, $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C})$ where the covariance matrix $\boldsymbol{C} = \boldsymbol{\Sigma} + \boldsymbol{A}\boldsymbol{A}^T$. Due to the diagonality of $\boldsymbol{\Sigma}$, the observed variables $\boldsymbol{x}$ are conditionally independent given the values of the latent variables $\boldsymbol{s}$. Thus the reduced-dimensional distribution $\boldsymbol{s}$ is intended to model the dependencies between the observed variables. This is in contrast to PCA which treats the inter-variable dependencies and the independent noise being identical. Factor analysis seeks for a factor loading matrix which best model the covariance structure of the observation data. In general, the columns of the factor loading matrix do not correspond to the principal subspace of the data. Maximum likelihood solution to factor analysis can be found in [7].

3. PROBABILISTIC PCA

This section reviews PPCA [10] for the case of isotropic Gaussian noise model and zero noise limit. In the limit of zero noise, PPCA is known as EM-PCA [5].

3.1. Isotropic Gaussian Noise

In general factor loadings $\boldsymbol{A}$ differ from the principal axes due to the diagonal noise model $\boldsymbol{\Sigma}$. Principal components emerges when independent noise variables have common variance σ^2, i.e., noise is isotropic Gaussian. Recently Tipping and Bishop [10] showed that under an isotropic noise structure, the maximum likelihood estimator $\boldsymbol{A}_{ML}$ spanns a principal subspace (which consists of scaled and rotated principal eigenvectors of the sample covariance matrix $\boldsymbol{R}$) even when the covariance model is approximate.

We assume an isotropic noise model, $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$, which implies a probability distribution over data space for a given $\boldsymbol{s}$ given by

$$p(\boldsymbol{x}|\boldsymbol{s}) = \frac{1}{(2\pi\sigma^2)^{\frac{m}{2}}} \exp\left\{-\frac{1}{2\sigma^2}\|\boldsymbol{x} - \boldsymbol{A}\boldsymbol{s}\|^2\right\}. \tag{2}$$

A Gaussian prior over the latent variables is used, i.e.,

$$p(\boldsymbol{s}) = \frac{1}{(2\pi)^{\frac{n}{2}}} \exp\left\{-\frac{1}{2}\|\boldsymbol{s}\|^2\right\}. \tag{3}$$

The marginal distribution of the data has the form of

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{s})p(\boldsymbol{s})d\boldsymbol{s} = \frac{1}{(2\pi)^{\frac{m}{2}}|\boldsymbol{C}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\boldsymbol{s}^T\boldsymbol{C}^{-1}\boldsymbol{s}\right\}, \quad (4)$$

where the model covariance is

$$\boldsymbol{C} = \boldsymbol{A}\boldsymbol{A}^T + \sigma^2\boldsymbol{I}. \quad (5)$$

Then the log-likelihood of observing the data under this model is

$$\mathcal{L} = \sum_{t=1}^{N} \log p(\boldsymbol{x}_t|\theta) = -\frac{Nm}{2}\log(2\pi) - \frac{N}{2}\log|\boldsymbol{C}| - \frac{N}{2}\mathrm{tr}\left\{\boldsymbol{C}^{-1}\boldsymbol{R}\right\}, \quad (6)$$

where $\boldsymbol{R}$ is the sample covariance matrix given by

$$\boldsymbol{R} = \frac{1}{N}\sum_{t=1}^{N}\boldsymbol{x}_t\boldsymbol{x}_t^T. \quad (7)$$

It was shown in [10] that with $\boldsymbol{C}$ given by (5), the only non-zero stationary points of $\frac{\partial\mathcal{L}}{\partial\boldsymbol{A}}$ occur for

$$\boldsymbol{A} = \boldsymbol{U}_n\left(\boldsymbol{\Lambda}_n - \sigma^2\boldsymbol{I}\right)^{\frac{1}{2}}\boldsymbol{Q}, \quad (8)$$

where the n column vectors in $\boldsymbol{U}_n$ are eigenvectors of the sample covariance matrix $\boldsymbol{R}$, with corresponding eigenvalues in the diagonal matrix $\boldsymbol{\Lambda}_n$, and $\boldsymbol{Q}$ is an arbitrary $n \times n$ orthogonal rotation matrix.

Following Rubin and Thayer [7], Tipping and Bishop derived an EM algorithm for maximizing the log-likelihood (6). Here I briefly review the EM algorithm for probabilistic PCA. See [10] for more details.

In the framework of EM, the latent variables $\{\boldsymbol{s}_t\}$ are treated as *missing data*. The complete-data log-likelihood $\mathcal{L}_c$ is given by

$$\mathcal{L}_c = \sum_{t=1}^{N}\log p(\boldsymbol{x}_t, \boldsymbol{s}_t|\theta) = \sum_{t=1}^{N}\left[\text{const} - \frac{m}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\|\boldsymbol{x}_t - \boldsymbol{A}\boldsymbol{s}_t\|^2 - \frac{1}{2}\boldsymbol{s}_t^T\boldsymbol{s}_t\right]. \quad (9)$$

Algorithm Outline: PPCA

E-step Compute sufficient statistics

$$\langle\boldsymbol{s}_t\rangle = \boldsymbol{M}^{-1}\boldsymbol{A}^T\boldsymbol{x}_t, \quad (10)$$

$$\left\langle\boldsymbol{s}_t\boldsymbol{s}_t^T\right\rangle = \sigma^2\boldsymbol{M}^{-1} + <\boldsymbol{s}_t><\boldsymbol{s}_t^T>, \quad (11)$$

where $\boldsymbol{M} = \sigma^2\boldsymbol{I} + \boldsymbol{A}^T\boldsymbol{A}$.

M-step Re-estimate the parameters $\boldsymbol{A}$ and σ^2 by

$$\widehat{\boldsymbol{A}} = \left(\sum_{t=1}^{N}\boldsymbol{x}_t\left\langle\boldsymbol{s}_t^T\right\rangle\right)\left(\sum_{t=1}^{N}\left\langle\boldsymbol{s}_t\boldsymbol{s}_t^T\right\rangle\right)^{-1}, \quad (12)$$

$$\widehat{\sigma^2} = \frac{1}{m}\left\{\mathrm{tr}\left[\boldsymbol{R} - \boldsymbol{R}\widehat{\boldsymbol{A}}\boldsymbol{M}^{-1}\widehat{\boldsymbol{A}}^T\right]\right\}. \quad (13)$$

3.2. Zero Noise Limit

Probabilistic PCA algorithm described in previous section is able to find scaled and rotated principal eigenvectors of the sample covariance matrix of the observed variables. It was derived in the framework of factor analysis with isotropic Gaussian noise model. PCA is a limiting case of the linear Gaussian model [6] (which factor analysis is based on) as the covariance of the noise $\boldsymbol{v}$ becomes infinitesimally small and equal in all directions. Hence a simple EM algorithm for PCA can be obtained by taking the zero noise limit into account. In the zero noise limit ($\sigma^2 \to 0$), the likelihood of a data point $\boldsymbol{x}$ is dominated solely by the squared distance between it and its reconstruction $\boldsymbol{A}\boldsymbol{s}$. In such a case, the posterior collapses to a single point and the covariance becomes zero, i.e.,

$$p(\boldsymbol{s}|\boldsymbol{x}) = \mathcal{N}((\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{x}, 0) = \delta(\boldsymbol{s} - (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{x}). \quad (14)$$

Now inference reduces to simple least squares projection, which leads to a simple EM algorithm [5] that is summarized below.

Algorithm Outline: PPCA (zero noise limit)

E-step Inference is carried out by LS projection,

$$\boldsymbol{S} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{X}, \quad (15)$$

where

$$\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N], \quad \boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]. \quad (16)$$

M-step Re-estimate the matrix $\boldsymbol{A}$ by

$$\widehat{\boldsymbol{A}} = \boldsymbol{X}\boldsymbol{S}\left(\boldsymbol{S}\boldsymbol{S}^T\right)^{-1}. \quad (17)$$

4. OJA'S SUBSPACE RULE

Let us consider a linear feedforward neural networks whose output $\boldsymbol{y}_t \in \mathbb{R}^n$ is described by

$$\boldsymbol{y}_t = \boldsymbol{W}\boldsymbol{x}_t, \quad (18)$$

where $\boldsymbol{W} \in \mathbb{R}^{n\times m}$ is a synaptic weight matrix.

The Oja's subspace learning rule [4] is given by

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t + \eta_t\boldsymbol{y}_t\boldsymbol{x}_t^T\left\{\boldsymbol{I} - \boldsymbol{W}_t^T\boldsymbol{W}_t\right\} = \boldsymbol{W}_t + \eta_t\left\{\boldsymbol{y}_t\boldsymbol{x}_t^T - \boldsymbol{y}_t\boldsymbol{y}_t^T\boldsymbol{W}_t\right\}, \quad (19)$$

where $\eta_t > 0$ is the learning rate. For the case of $n = 1$, the subspace rule (19) becomes the well known Oja's algorithm [3]) which is known to extract the eigenvector associated the largest eigenvalue of the input data covariance matrix. When the convergence of (19) is achieved, the synaptic weight matrix $\boldsymbol{W}$ corresponds to the eigenspace which is spanned by the principal eigenvectors of the input data covariance matrix.

Besides the Oja's subspace rule, a variety of PCA neural networks have been developed (see [1] and references therein).

5. SEQUENTIAL EM FOR PSA

5.1. Separable Least Squares

The PPCA algorithm for the case of zero noise limit can be also derived in the framework of separable LS method. Moreover we can employ the sequential LS (also known as recursive LS) in order to develop an on-line algorithm which learns principal subspace of the observed variables.

As pointed out in [5], in the zero noise limit, the likelihood of a data point $\boldsymbol{x}$ is dominated solely by the squared distance between it and its reconstruction $\boldsymbol{As}$. In such a case, ML estimation of both $\boldsymbol{A}$ and $\boldsymbol{s}$ becomes a separable LS minimization problem. The LS estimates, $\boldsymbol{A}$ and $\boldsymbol{S}$ are computed by

$$\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{S}} = \min_{\boldsymbol{A},\boldsymbol{S}} \|\boldsymbol{X} - \boldsymbol{AS}\|_F^2 . \tag{20}$$

The separable LS minimization is carried out in two steps. First we minimize (20) with respect to $\boldsymbol{A}$ with $\boldsymbol{S}$ being fixed. It leads to

$$\widehat{\boldsymbol{A}} = \boldsymbol{XS}\left(\boldsymbol{SS}^T\right)^{-1} \tag{21}$$

which corresponds to the M-step in PPCA (for the case of zero noise limit).

The estimate $\widehat{\boldsymbol{A}}$ is substituted back into (20), then we obtain a new criterion which is a function of $\boldsymbol{S}$ only

$$\min_{\boldsymbol{S}} \left\| \boldsymbol{X}\boldsymbol{P}_{\boldsymbol{S}}^{\perp} \right\|_F^2 , \tag{22}$$

where $\boldsymbol{P}_{\boldsymbol{S}}^{\perp}$ is the orthogonal projection matrix given by

$$\boldsymbol{P}_{\boldsymbol{S}}^{\perp} = \boldsymbol{I} - \boldsymbol{S}^T\left(\boldsymbol{SS}^T\right)^{-1}\boldsymbol{S}. \tag{23}$$

The minimization of (22) is achieved when

$$\boldsymbol{S} = \left(\widehat{\boldsymbol{A}}^T\widehat{\boldsymbol{A}}\right)^{-1}\widehat{\boldsymbol{A}}^T\boldsymbol{X}, \tag{24}$$

which correspond to the E-step.

5.2. Sequential LS

For sequential estimation of $\boldsymbol{A}$ and $\boldsymbol{s}$, we consider the weighted LS minimization problem where the objective function is given by

$$\mathcal{E} = \sum_{k=1}^{t} \beta^{t-k} \|\boldsymbol{x}_k - \boldsymbol{As}_k\|^2 , \tag{25}$$

where $0 < \beta \leq 1$ is the forgetting factor.

Our objective is to compute $\boldsymbol{A}_t$ and $\boldsymbol{s}_t$, assuming a good estimate of $\boldsymbol{s}_{t-1}$ (or equivalently $\boldsymbol{A}_{t-1}$) is available. The exponential weighting is used to de-emphasize old data in a time-varying environment. Setting the derivative of $\mathcal{E}$ with respect to $\boldsymbol{A}$ to be zero, then we have

$$\boldsymbol{A}_t = \boldsymbol{R}_{\boldsymbol{xs},t}\left[\boldsymbol{R}_{\boldsymbol{ss},t}\right]^{-1}, \tag{26}$$

where

$$\begin{aligned} \boldsymbol{R}_{\boldsymbol{xs},t} &= \sum_{k=1}^{t} \beta^{t-k}\boldsymbol{x}_k\boldsymbol{s}_k^T, \\ \boldsymbol{R}_{\boldsymbol{ss},t} &= \sum_{k=1}^{t} \beta^{t-k}\boldsymbol{s}_k\boldsymbol{s}_k^T. \end{aligned} \tag{27}$$

Define $\boldsymbol{P}_t = \boldsymbol{R}_{\boldsymbol{ss},t}^{-1}$ and apply the matrix inversion lemma. Then we have a recursion equation for updating $\boldsymbol{P}_t$

$$\boldsymbol{P}_t = \frac{1}{\beta}\left\{\boldsymbol{P}_{t-1} - \frac{\boldsymbol{P}_{t-1}\boldsymbol{s}_t\boldsymbol{s}_t^T\boldsymbol{P}_{t-1}}{\beta + \boldsymbol{s}_t^T\boldsymbol{P}_{t-1}\boldsymbol{s}_t}\right\}. \tag{28}$$

Using the recursion (28), the adaptation for $\boldsymbol{A}$ is given by

$$\begin{aligned} \boldsymbol{A}_t &= \boldsymbol{R}_{\boldsymbol{xs},t}\boldsymbol{P}_t \\ &= \boldsymbol{A}_{t-1} + [\boldsymbol{x}_t - \boldsymbol{A}_{t-1}\boldsymbol{s}_t]\frac{\boldsymbol{s}_t^T\boldsymbol{P}_{t-1}}{\beta + \boldsymbol{s}_t^T\boldsymbol{P}_{t-1}\boldsymbol{s}_t}. \end{aligned} \tag{29}$$

Algorithm Outline: Sequential EM

E-step Estimate $\boldsymbol{s}_t$ by the LS projection

$$\boldsymbol{s}_t = \left(\boldsymbol{A}_{t-1}^T\boldsymbol{A}_{t-1}\right)^{-1}\boldsymbol{A}_{t-1}^T\boldsymbol{x}_t. \tag{30}$$

M-step Estimate $\boldsymbol{A}_t$ by

$$\boldsymbol{A}_t = \boldsymbol{A}_{t-1} + \epsilon_t\frac{\boldsymbol{s}_t^T\boldsymbol{P}_{t-1}}{\beta + \boldsymbol{s}_t^T\boldsymbol{P}_{t-1}\boldsymbol{s}_t}, \tag{31}$$

where

$$\boldsymbol{\epsilon}_t = \boldsymbol{x}_t - \boldsymbol{A}_{t-1}\boldsymbol{s}_t, \tag{32}$$

6. A NUMERICAL EXAMPLE

A simple numerical example is given here to confirm fast convergence and high performance of the proposed algorithm. The algorithm is compared with the Oja's subspace rule [4]. The 3-dimensional observation vector $\boldsymbol{x}$ was generated with its covariance matrix given by

$$\begin{bmatrix} 1.391 & 0.173 & -0.536 \\ 0.173 & 0.032 & -0.078 \\ -0.536 & -0.078 & 2.584 \end{bmatrix}. \tag{33}$$

As a performance measure, I use the subspace error (SE) which is defined by

$$\mathrm{SE} = \frac{1}{\sqrt{n}}\left\|\boldsymbol{P}_{\boldsymbol{A}}^{\perp}\boldsymbol{U}_n\boldsymbol{U}_n^T\right\|, \tag{34}$$

where $\boldsymbol{P}_{\boldsymbol{A}}^{\perp}$ is the projection matrix onto the orthogonal subspace, i.e.,

$$\boldsymbol{P}_{\boldsymbol{A}}^{\perp} = \boldsymbol{I} - \boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T, \tag{35}$$

which is decided by the estimated values of $\boldsymbol{A}$ and $\boldsymbol{U}_n\boldsymbol{U}_n^T$ is the projection matrix onto the signal subspace that is computed from the SVD of the covariance matrix of the input data. When the estimated $\boldsymbol{A}$ spanns the true signal subspace, $\boldsymbol{P}_{\boldsymbol{A}}^{\perp}$ should orthogonal to $\boldsymbol{U}_n\boldsymbol{U}_n^T$. Hence the SE defined above is able to serve as a performance measure.

The matrices $\boldsymbol{A}_0$ (for the sequential EM) and $\boldsymbol{W}_0$ (for Oja's subspace rule) were initialized as a random matrix whose elements are drawn from uniformly distributed random variables over [0,1]. The learning rate in the Oja's subspace rule was $\eta_t = .01$. In Fig. 1, one can observe that the sequential EM algorithm converges to a solution much faster than the Oja's subspace rule and even after convergence, the former shows slight better performance than the latter.

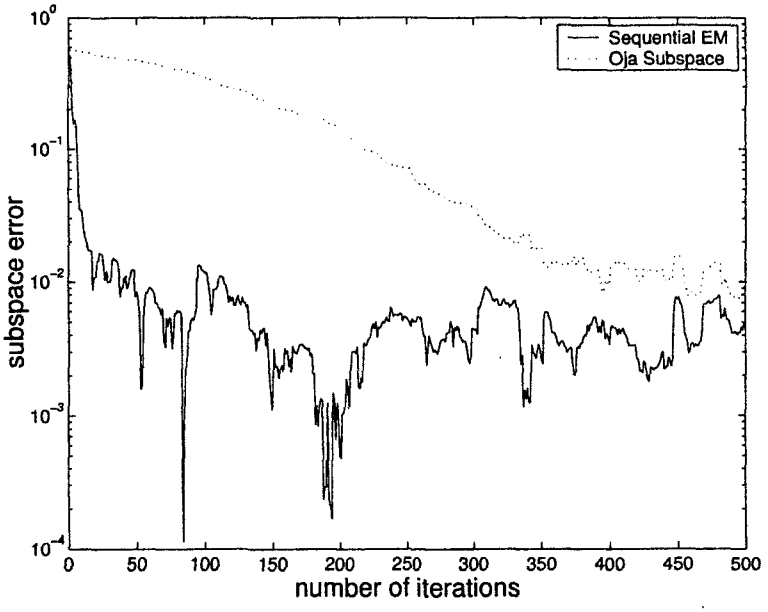

Figure 1: Convergence comparison for sequential EM algorithm and Oja's subspace rule in terms of subspace error.

7. CONCLUSION

I have described some relations between the EM algorithm and separable LS fitting in PPCA with zero noise limit. Based on this observation, I have developed a sequential EM learning for subspace analysis which is nothing but a sequential LS algorithm. The sequential EM algorithm can be easily extended to mixtures of PCA, which is currently under investigation.

8. ACKNOWLEDGMENT

This work was supported by KOSEF 2000-2-20500-009-5 and by Korea Ministry of Science and Technology under Brain Science and Engineering Research Program and International Cooperative Research Program and by Ministry of Education of Korea for its financial support toward the Electrical and Computer Engineering Division at POSTECH through its BK21 program.

9. REFERENCES

[1] K. I. Diamantaras and S. Y. Kung. *Principal Component Neural Networks: Theory and Applications*. John Wiely & Sons, INC, 1996.

[2] Z. Ghahramani and G. E. Hilton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, Department of Computer Science, University of Toronto, 1997.

[3] E. Oja. A simplified neuron model as a principal component analyzer. *J. Mathematical Biogology*, 15:267–273, 1982.

[4] E. Oja. Neural networks, principal component analysis, and subspaces. *International Journal of Neural Systems*, 1:61–68, 1989.

[5] S. Roweis. EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems*, volume 10, pages 626–632. MIT press, 1998.

[6] S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11(2):305–345, 1999.

[7] D. Rubin and D. Thayer. EM algorithms for ML factor analysis. *Psychometrika*, 47:69–76, 1982.

[8] L. Sirovich and M. Kirby. Low-dimensional procedure for the charaterization of human faces. *Journal of the Optical Society of America A*, 4(3):519–524, 1987.

[9] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

[10] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society B*, 61(3):611–622, 1999.

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